# DashBoard Control, Monitoring, and Workflow Automation System User Guide

**Version 6.2**

www.rossvideo.com

ROSS
Production Technology Experts

# Thank you for choosing Ross

You've made a great choice. We expect you will be very happy with your purchase of Ross Technology.

Our mission is to:

1. Provide a Superior Customer Experience
    - offer the best product quality and support
2. Make Cool Practical Technology
    - develop great products that customers love

Ross has become well known for the Ross Video Code of Ethics. It guides our interactions and empowers our employees. I hope you enjoy reading it below.

If anything at all with your Ross experience does not live up to your expectations be sure to reach out to us at *solutions@rossvideo.com*.

David Ross
CEO, Ross Video
*dross@rossvideo.com*

## Ross Video Code of Ethics

Any company is the sum total of the people that make things happen. At Ross, our employees are a special group. Our employees truly care about doing a great job and delivering a high quality customer experience every day. This code of ethics hangs on the wall of all Ross Video locations to guide our behavior:

1. We will always act in our customers' best interest.
2. We will do our best to understand our customers' requirements.
3. We will not ship crap.
4. We will be great to work with.
5. We will do something extra for our customers, as an apology, when something big goes wrong and it's our fault.
6. We will keep our promises.
7. We will treat the competition with respect.
8. We will cooperate with and help other friendly companies.
9. We will go above and beyond in times of crisis. *If there's no one to authorize the required action in times of company or customer crisis - do what you know in your heart is right. (You may rent helicopters if necessary.)*

# DashBoard User Guide


- Ross Part Number: **8351DR-004-6.2**
- Publication Date: September 2, 2014. Printed in Canada.
- Software Issue: **6.2**


The information contained in this Guide is subject to change without notice or obligation.

## Copyright


© 2014 Ross Video Limited. Ross® and any related marks are trademarks or registered trademarks of Ross Video Limited. All other trademarks are the property of their respective companies. PATENTS ISSUED and PENDING. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, mechanical, photocopying, recording, or otherwise, without the prior written permission of Ross Video. While every precaution has been taken in the preparation of this document, Ross Video assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.


## Patents

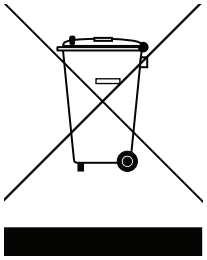
Ross Video products are protected by patent numbers US 7,034,886; US 7,508,455; US 7,602,446; US 7,802,802 B2; US 7,834,886; US 7,914,332; US 8,307,284; US 8,407,374 B2; US 8,499,019 B2; US 8,519,949 B2; US 8,743,292 B2; GB 2,419,119 B; GB 2,447,380 B. Other patents pending.

## Environmental Information

The equipment that you purchased required the extraction and use of natural resources for its production. It may contain hazardous substances that could impact health and the environment.

To avoid the potential release of those substances into the environment and to diminish the need for the extraction of natural resources, Ross Video encourages you to use the appropriate take-back systems. These systems will reuse or recycle most of the materials from your end-of-life equipment in an environmentally friendly and health conscious manner.

The crossed-out wheeled bin symbol invites you to use these systems.



If you need more information on the collection, reuse, and recycling systems, please contact your local or regional waste administration.

You can also contact Ross Video for more information on the environmental performances of our products.

# Company Address


**Ross Video Limited**
8 John Street
Iroquois, Ontario
Canada, K0E 1K0

**Ross Video Incorporated**
P.O. Box 880
Ogdensburg, New York
USA 13669-0880

General Business Office: (+1) 613 • 652 • 4886
Fax: (+1) 613 • 652 • 4425

Technical Support: (+1) 613 • 652 • 4886
After Hours Emergency: (+1) 613 • 349 • 0006

E-mail (Technical Support): techsupport@rossvideo.com
E-mail (General Information): solutions@rossvideo.com
Website: http://www.rossvideo.com

# Contents


## Introduction 1

Overview ....1-1
Features ....1-2
Documentation Terms ....1-2
Documentation Conventions ....1-2
Interface Elements ....1-2
User Entered Text ....1-2
Referenced Guides ....1-3
Menu Sequences ....1-3
Interface Navigation ....1-3
Important Instructions ....1-3
Getting Help ....1-3
Contacting Technical Support ....1-3

## Installing DashBoard 2

Before You Begin ....2-1
System Requirements ....2-1
Installing and Removing the DashBoard Control System Client ....2-2
Creating a Backup of the Settings and Licensed Features ....2-2
Installing on a Computer Running Microsoft® Windows® ....2-2
Installing on a Computer Running Apple® Mac® OS® X ....2-3
Installing on a Computer Running Linux® Fedora® ....2-3
Removing DashBoard ....2-3
Installing DashBoard Add-on Programs ....2-4
Getting Started ....2-4
Viewing Installation Details ....2-5
DashBoard Installation Details Overview ....2-5
Viewing Error Logs ....2-6

## Managing openGear Frames in DashBoard 3

Managing openGear Frames in DashBoard ....3-1
Adding openGear Frames to DashBoard ....3-1
Re-naming an openGear Frame in the Tree View ....3-2
Removing openGear Frames from a Tree View ....3-2

## Using the DashBoard Interface 4

DashBoard Interface Overview ....4-1
Status Indicators ....4-2
DashBoard Basic Tree View ....4-3
Overview ....4-3
Using the Basic Tree View ....4-5
Using the Advanced Tree View ....4-6
Overview ....4-6
Using the Advanced Tree View ....4-8
The Device Editor Area ....4-8
Overview ....4-8
Using the Device Editor Feature ....4-10
Using Layouts ....4-10
Overview ....4-10
Managing Your Layouts ....4-11

Keyboard Shortcuts ..... 4-12
Overview ..... 4-13
Managing the Keyboard Shortcuts ..... 4-14
Resolving Conflicts ..... 4-15
Importing and Exporting Libraries ..... 4-15
Resetting to Default Values ..... 4-16
Restoring Keyboard Shortcut Commands for Copy, Cut, Paste, Undo, and Redo ..... 4-16
Using DashBoard Help ..... 4-16
Configuring the DashBoard Help Display Options ..... 4-17
Importing openGear Help ..... 4-18
Preferences ..... 4-19
Secure Storage ..... 4-19
Web Browser ..... 4-19

PanelBuilder™ 5

About PanelBuilder ..... 5-1
CustomPanel Examples ..... 5-2
PanelBuilder Concepts and Terminology ..... 5-4
Creating a CustomPanel ..... 5-8
Edit Mode ..... 5-9
The Edit Mode Toolbar ..... 5-10
Adding Device Editors, Device Summaries, and Device Controls ..... 5-15
Device Editors and Device Summaries ..... 5-15
Device Controls ..... 5-16
Adding Basic Components ..... 5-18
Blank Canvases ..... 5-20
Tab Groups ..... 5-21
Image Canvases ..... 5-22
Split Panels ..... 5-23
Tables ..... 5-25
Simple Grids ..... 5-26
Flow Containers (Wrap Content) ..... 5-27
Labels ..... 5-28
Links to Device Editors or Other CustomPanels ..... 5-29
Buttons ..... 5-30
Line Segments ..... 5-32
Web Browser Instances ..... 5-34
Adding Data-Backed Components ..... 5-34
Parameter Displays ..... 5-35
Data-Backed Labels ..... 5-36
Editable Text Areas ..... 5-36
Option Choice Controls ..... 5-37
Numeric Choice Controls ..... 5-37
Toggle Choice Controls ..... 5-38
Creating a Row, Column, or Grid of Data-Backed Buttons ..... 5-39
Timers ..... 5-40
Creating a Timer ..... 5-41
Adding Timer Labels and Timer Control Buttons to a CustomPanel ..... 5-41
Timer Control Functions ..... 5-42
Assigning Tasks to Buttons, Labels, and Timers ..... 5-43
Assigning ogScript Tasks ..... 5-43
Assigning Pauses ..... 5-44
Assigning RossTalk Commands ..... 5-44
Assigning CamBot Commands ..... 5-45
Assigning VDCP Commands ..... 5-46
Assigning PBUS Commands ..... 5-47

Using the Global List ....5-48
Assigning Data Modification Tasks ....5-48
Assigning Timer Control Tasks ....5-49
Editing a Task ....5-50
Triggering Tasks Externally ....5-50
Creating a GPI Trigger ....5-50
Triggering Tasks Using Keyboard Shortcuts ....5-52
Triggering Tasks Using RossTalk Messages ....5-52
Triggering Tasks Using the ogscript.fireGPI Function ....5-53
Editing Components ....5-53
Component Tree ....5-55
Attribute Editor Tabs ....5-56
Abs Attributes Tab ....5-58
Blank Attributes Tab ....5-59
Browser Attributes Tab ....5-59
Button Attributes Tab ....5-60
Color Attributes Tab ....5-60
Container Attributes Tab ....5-61
Dropspot Attributes Tab ....5-62
Editor Attributes Tab ....5-62
Flow Attributes Tab ....5-63
Label Attributes Tab ....5-64
Listener Attributes Tab ....5-64
Lookup Attributes Tab ....5-66
Meta Attributes Tab ....5-66
ogScript Attributes Tab ....5-66
Param Attributes Tab ....5-68
Position/Stretch Attributes Tab ....5-70
Simplegrid Attributes Tab ....5-71
Source Tab ....5-72
Split Attributes Tab ....5-73
Statuscombo Attributes Tab ....5-73
Style Tab ....5-74
Tab Attributes Tab ....5-77
Table Attributes Tab ....5-78
Table Cell Attributes Tab ....5-79
Tag Attributes Tab ....5-80
Timer Attributes Tab ....5-81
Timertask Attributes Tab ....5-82
Tr Attributes Tab ....5-82
TreeElement Attributes Tab ....5-82
Anchor Points and Background Alignment ....5-83
Locking Panel Proportions ....5-86
Parameters and Data Sources ....5-87
The Add/Edit Parameter Window ....5-87
Creating a New Parameter ....5-88
Associating a Data Source with a CustomPanel ....5-90
Router Control Panels ....5-91

DataSafe™ 6

DataSafe Overview ....6-1
DataSafe Basics ....6-2
Saving a DataSafe File ....6-3
Restoring Configurations to Devices ....6-3
Notes on Saving and Restoring Parameters ....6-5
Forcing DataSafe Updates ....6-5

Configuring Devices 7
Configuring Online Devices in DashBoard ..... 7-1
Configuring Devices in DashBoard ..... 7-1
Re-naming an openGear Slot in the **Tree View** ..... 7-2
Automatic Discovery ..... 7-2
Troubleshooting ..... 7-3
Removing Devices from the Tree View ..... 7-3
Using the File Navigator ..... 7-3
Upgrading Device Software ..... 7-4
Upgrading Device Software ..... 7-4
Troubleshooting the Software Upload Process ..... 7-5

# Introduction

This chapter contains the following sections:

- Overview
- Documentation Terms
- Documentation Conventions
- Getting Help

## Overview

This guide provides an overview of installing, setting up, and using the DashBoard client. For information on using the DashBoard Server and User Rights Management features, refer to the ***DashBoard Server and User Rights Management User's Guide***.

The DashBoard Control System is built on Ethernet and TCP/IP technology, which allows remote access across LAN architectures. DashBoard offers the ability to view multiple frames with full control and alarming of all populated slots inside an openGear frame. This simplifies the setup of numerous devices in a large installation and offers the ability to centralize monitoring. The devices define their controllable parameters and layout to DashBoard, so the control interface is always up-to-date.

Alarms raised by devices in the frame are reported at the uppermost level, making it quick and easy to identify potential failures or problems.

#### Monitoring

A network of DashBoard Connect compatible devices can be monitored, allowing users to quickly isolate and correct potential problems from a central monitoring station.

#### Control

DashBoard offers real-time control of DashBoard Connect compatible devices. Parameter items and menus vary depending on device functionality.

#### Workflow Automation

DashBoard's PanelBuilder™ enables you to create customized user interfaces to control a wide range of Ross Video products and third-party systems. PanelBuilder includes a rich set of drag-and-drop tools to make creation of CustomPanel interfaces easy. CustomPanels can be tailored to suit the exact requirements and preferences of each operator. You can further enhance your CustomPanels using Ross Video ogScript functions, to create user interfaces that perfectly match your desired workflow.

#### DataSafe™

DataSafe enables you to back up and restore device parameters to and from a single file, and to copy parameters from one device to a another device or even a group of devices via DashBoard. This feature allows you to hot-swap devices while retaining configurations, and DashBoard is not required to have the parameters on the new device updated. You can update a subset of devices instead of the entire connected view or to replace a device and have the previous device settings automatically loaded onto the new device. When using openGear cards, this feature is available for frames using the Network Controller Cards series only.

#### Software Upgrades

You can upgrade the software and firmware on devices, such as openGear cards, in the field using DashBoard. The upgrade utility verifies firmware and software upgrades against device hardware and prevents accidental loading of incorrect files to the wrong hardware. DashBoard is designed to allow for future feature plug-ins and upgrades to allow users the ability to customize control and monitoring needs.

### Features

The following features make DashBoard a unique system for your openGear requirements:

- Automatic discovery of openGear and DashBoard Connect devices
- Access multiple openGear frames on a single control network
- Access multiple DashBoard applications on a single control network
- Ability to have multiple control windows active and available on one screen
- Store and upload configurations to multiple devices (using DataSafe)
- Ability to perform batch software upgrades to allow multiple cards, of the same model, to be upgraded at one time from any DashBoard terminal on the network
- Customize views with the Advanced Tree View feature
- Ability to create customized monitoring and control interfaces using DashBoard PanelBuilder
- Extensible plug-in architecture
- DashBoard software and documentation are available for download from the Ross Video website
- Java™ based for installation in Microsoft® Windows®, Mac® OS, and Linux® Fedora®
- Software and firmware upgrades via ethernet

## Documentation Terms

- All references to the **DFR-8300 series frame** also includes all version of the 10-slot and 20-slot frames and any available options.
- "**Card**" refers to openGear terminal devices within openGear frames, including all components and switches.
- "**DashBoard window**" refers to the main DashBoard client interface.
- "**Device**" refers to a product that can be monitored and controlled using DashBoard. Devices include NK routers, BlackStorm servers, openGear cards, and DashBoard Connect devices.
- "**Frame**" refers to any openGear frame within your video system.
- "**Network Controller Card**" refers to the MFC-8310-N series and MFC-8320-N series cards unless otherwise noted.
- "**System**" refers to the mix of interconnected production and terminal equipment in your environment.
- "**Tree View**" refers to the Basic Tree View and Advanced Tree View unless otherwise noted.

"**User**" refers to the person who uses the DashBoard client.

## Documentation Conventions

Special text formats are used in this guide to identify parts of the user interface, text that a user must enter, or a sequence of menus and submenus that must be followed to reach a particular command.

### Interface Elements

Bold text is used to identify a user interface element such as a dialog box, menu item, or button. For example:

In the **Media Manager Client**, click **Channel 1** the **Channels** section.

### User Entered Text

Courier text is used to identify text that a user must enter. For example:

In the **File Name** box, enter `Channel01.property`.

### Referenced Guides

Italic text is used to identify the titles of referenced guides, manuals, or documents. For example:

***DashBoard Server and User Rights Management User's Guide***

### Menu Sequences

Menu arrows are used in procedures to identify a sequence of menu items that you must follow. For example, if a step reads "**Server** > **Save As**," you would click the **Server** menu and then click **Save As**.

### Interface Navigation

Navigation procedures assume that you are running Microsoft® Windows®. If you are running Mac® OS or Linux® Fedora®, menu names and options may differ.

### Important Instructions

Star icons are used to identify important instructions or features. For example:

✶ Contact your I.T. Department if you experience communication issues with DashBoard and are running anti-virus software.

## Getting Help

To access the built-in Help system, click **Help** in the main toolbar.

Alternatively a user can press **F1** to open **Dynamic Help**. The user can then click on areas of the window to display corresponding help information.

The DashBoard User Guide is also supplied as a print-ready PDF file on the Ross Video website.

### Contacting Technical Support

At Ross Video, we take pride in the quality of our products, but if problems occur, help is as close as the nearest telephone.

Our 24-hour Hot Line service ensures you have access to technical expertise around the clock. After-sales service and technical support is provided directly by Ross Video personnel. During business hours (Eastern Time), technical support personnel are available by telephone. After hours and on weekends, a direct emergency technical support phone line is available. If the technical support person who is on call does not answer this line immediately, a voice message can be left and the call will be returned shortly. This team of highly trained staff is available to react to any problem and to do whatever is necessary to ensure customer satisfaction.

- **Technical Support:** (+1) 613-652-4886
- **After Hours Emergency:** (+1) 613-349-0006
- **E-mail:** techsupport@rossvideo.com
- **Website:** http://www.rossvideo.com

# Installing DashBoard

This chapter provides instructions for installing the DashBoard Control System client software. For information on installing the DashBoard URM and Server, refer to the ***DashBoard Server and User Rights Management User Manual***.

The following topics are discussed:

- Before You Begin
- Installing and Removing the DashBoard Control System Client
- Getting Started
- Viewing Installation Details

## Before You Begin

Before installing any software for your DashBoard client, ensure that you exit all other programs currently running.

✶ Contact your I.T. Department if you experience communication issues with DashBoard and are running anti-virus software. You may need to verify that there is an exception in your firewall to allow DashBoard to receive TCP data via Port 5353.

### System Requirements

Refer to the following sections for information on the system requirements for DashBoard.

#### Microsoft® Windows® 8 or Windows® 7 Systems

The following are the minimum requirements when installing DashBoard on a Microsoft® Windows® system:

- 32bit or 64bit configurations
- Intel® Pentium 4, 1.6GHz (Intel® Core™ 2 Duo recommended)
- Minimum 2GB RAM (4GB or more recommended)
- 300MB available in HD space
- Microsoft® Internet Explorer® version 8 (minimum)

**Note:** If you are running DashBoard on a Microsoft® Surface Pro®, you must use a physical keyboard or enable the standard keyboard layout. For more information, see "**Installing on a Computer Running Microsoft® Windows®**" on page 2–2.

#### Apple Mac® OS X 10.7.3 (or higher) Systems (64bit)

The following are the minimum requirements when installing DashBoard on a Mac® OS system:

- 64bit configuration only
- Intel® processor
- Minimum 2GB RAM (4GB or more recommended)
- 300MB available HD space
- Mac® Safari®

#### Linux® Fedora® 18 (or higher) Systems

The following are the minimum requirements when installing DashBoard on a Linux® system:

- 32bit configuration only (If you are running 64bit, ensure to install the 32bit libraries)
- Intel® Pentium 4, 1.6GHz (Intel® Core™ 2 Duo recommended)
- Minimum 2GB RAM (4GB or more recommended)
- 300MB available in HD space
- Mozilla 1.4 GTK2 - 1.7.x GTK2, XULRunner 1.8.x - 1.9.x, 3.6.x and 10.x (but not 2.x nor 4.x - 9.x), WebKitGTK+ 1.2.x and newer
- Version 2.2.1 GTK+ widget toolkit and associated libraries (GLib, Pango) are required

## Installing and Removing the DashBoard Control System Client

This section includes instructions on installing the DashBoard client to your computer. The DashBoard software and user guide are available from the Ross Video website.

### Creating a Backup of the Settings and Licensed Features

Regular backups of your DashBoard settings is recommended, or if you want to create a backup of your DashBoard settings and license files, before installing a new version of DashBoard.

- **Microsoft® Windows®** — DashBoard automatically uninstalls a previously installed version, but not your settings.
  - › To create a back up of your settings for Microsoft® XP® and earlier, navigate to the metadata folder. Depending on your operating system version, this folder may be located at **c:\Program Files\DashBoard\workspace**. Copy the workspace folder contents to a new location.
  - › If you are running Microsoft® Vista® or higher, the metadata folder is at: **c:\DashBoard\workspace**. Copy the folder to a new location.
- **Apple® Mac® OS®** — DashBoard saves application information to the following location on your computer: **/Library/Application Support/openGear/DashBoard**. To create a backup of your settings, copy the **workspace** folder to a different location on your system.
- **Linux® Fedora®** — DashBoard creates a **workspace** directory inside of the current DashBoard directory when the application is launched. To create a backup copy of your settings, copy the workspace directory to a different location on your system.

### Installing on a Computer Running Microsoft® Windows®

The Install Wizard automatically uninstalls any previous software versions before proceeding. The Dashboard main folder is located at **c:\DashBoard**.

#### To install DashBoard on a computer running Microsoft® Windows®:

1. Access the DashBoard software using one of the following methods:
   - Navigate to the Ross Video website and download the current software file.
   - Load the DashBoard software CD into the DVD/CD ROM tray of your computer.
2. If you are accessing the software from a CD, the **Installation Wizard** automatically runs. If the Wizard does not automatically run, you can also install the DashBoard software, navigate to your DVD/CD ROM drive in the **Navigation Pane**, so that the CD contents are displayed in the **Main Window** of Windows Explorer.
3. Launch **DBx.x.x-Win-setup.exe** to begin installing the DashBoard program onto your computer.
4. Follow the prompts to complete the installation of DashBoard onto your computer.
5. If you are running DashBoard on a Microsoft® Surface Pro®, you must use a physical keyboard or enable the standard keyboard layout.

This is because the default virtual keyboards do not contain function keys necessary to use DashBoard's keyboard shortcuts.

To enable the standard keyboard layout:

a. In **PC settings**, tap **General**.

b. Under **Touch keyboard** options, turn on the **Make the standard keyboard layout available** option.

## Installing on a Computer Running Apple® Mac® OS® X

If you have a previous version of DashBoard installed, it is recommended that you remove the DashBoard directory from the **Applications** folder before proceeding.

**To install DashBoard on a computer running Apple® Mac® OS® X:**

1. Access the DashBoard software using one of the following methods:
   - Navigate to the Ross Video website and click the link to the DashBoard installation program for Apple OS X.
   - Load the DashBoard software CD into the DVD/CD ROM tray of your computer and navigate to the DashBoard *.pkg file for Apple OS X.
2. Navigate to the *.pkg file specified in step 1.
3. Right-click the *.pkg file and click **Open**.

   A message appears about being unable to open the app because it is from an unidentified developer.
4. In the message box, click **Open**.
5. Follow the prompts to complete the installation of DashBoard onto your computer.

## Installing on a Computer Running Linux® Fedora®

If you have a previous version of DashBoard installed and want to keep your settings, remove all folders, except the workspace directory, from the DashBoard directory. Note that the uninstall function of DashBoard may delete the directory that also contains your user data.

**To install DashBoard on a computer running Linux® Fedora®:**

1. Access the DashBoard software using one of the following methods:
   - Navigate to the Ross Video website and click the link to the **DashBoard.tar.gz file** for Linux.
   - Load the DashBoard software CD into the DVD/CD ROM tray of your computer.
2. Extract the new **DashBoard.tar.gz file** to your system. Note that this file contains a top-level directory called DashBoard.
3. To run DashBoard, click the **DashBoard** icon in the top level of the DashBoard program folder.

   If you run DashBoard from the command line, ensure that you switch to the main DashBoard directory before launching DashBoard.

✶ DashBoard creates a workspace directory where the application is launched. This directory includes saved settings and licenses for optional features. To provide consistent retrieval of saved settings and program features, Ross Video recommends that you always launch Dashboard from within the same directory.

## Removing DashBoard

Create a backup for your settings and licensed features before removing Dashboard if you wish to retain your user data. Refer to the section "**Creating a Backup of the Settings and Licensed Features**" on page 2-2.

**To remove DashBoard from a computer running Microsoft® Windows®:**

- Use the **Add/Remove Software** program located in the Windows® Control Panel.

✶ Deleting the DashBoard directory before running the **Add/Remove Software** program results in a number of dead registry and Start Menu items on your system. You can delete the directory to remove your user data after you run the **Add/Remove Software** program.

**To remove DashBoard from a computer running Apple® OS X®:**

1. Remove your user data from the application support library directory.
2. Delete the Dashboard folder from your computer.

**To remove DashBoard from a computer running Linux® Fedora®:**

- Delete the Dashboard folder from your computer.

## Installing DashBoard Add-on Programs

This section briefly outlines how to install Add-on programs for DashBoard such as:

- **openGear Extra Feature Pack** — If you have an executable openGear Extra Feature pack, and are running Microsoft® Windows®, use the following procedure to install your feature pack. If you are not running Microsoft® Windows®, refer to the feature pack documentation for installation details.
- **Unicode Font Support Pack** — This program installs an international character set for DashBoard and provides access to fonts capable of displaying all Unicode characters. This program is not available for systems running Apple Mac® OS®. This option may be required if you are using an openGear device designed with fonts not supported by the base Java font system. Consult your device documentation for requirements and details.

**To install a DashBoard Add-on program:**

1. Navigate to the openGear website (www.opengear.tv) and download the required add-on program for your system.
2. If you are running Microsoft® Windows®:
   - Launch the **Setup Wizard** for your Add-on program by selecting the corresponding ***.exe file**.
   - Follow the prompts to complete the installation of the Add-on program.
3. If you are running Linux® Fedora®:
   - Ensure you are in the main **DashBoard** directory before extracting the Add-on program.
   - Copy the required ***.zip** to the main **DashBoard** directory.
   - Extract the ***.zip** file.

# Getting Started

This section provides a brief introduction to launching the DashBoard client, and accessing some of its features.

**To launch DashBoard:**

1. Ensure that you have installed the DashBoard software as outlined in the section "**Installing and Removing the DashBoard Control System Client**" on page 2–2.
2. Launch the DashBoard client by double-clicking its icon on your desktop.
3. If you are using DashBoard Server and URM, you will be prompted for a user name and password.

**Using the Full-screen mode**

You can set the DashBoard interface to full-screen by performing one of the following:

- Press **Shift** + **F11**.
- Select **Window** > **Full Screen** from the main DashBoard toolbar.

To exit out of full-screen mode by performing one of the following:

- Press **Shift** + **F11**.
- Right-click the DashBoard icon in the system tray and unselect **Full Screen**.

#### Displaying multiple DashBoard windows

When operating in a multi-screen environment, you can open multiple DashBoard windows as follows:

- Select **Window** > **New Window** from the main DashBoard menu bar.

#### Locking the DashBoard window

You can lock the DashBoard window, preventing users from accessing the DashBoard client running on your computer by performing one of the following:

- Press **Shift** + **F4**.
- Select **Window** > **Lock Screen**.

To unlock the interface:

- Use the provided **Unlock** wheel.

## Viewing Installation Details

You can view your installation history, activities in DashBoard, and error logs using the **DashBoard Installation Details** dialog box available from the **About DashBoard** dialog box.

### DashBoard Installation Details Overview

This section briefly explains the components of the **DashBoard Installation Details** dialog box should you need to view it for troubleshooting purposes or as directed by Ross Video Technical Support.

The **DashBoard Installation Details** dialog box includes the following tabs:

- **Installed Software** tab — This tab displays the currently installed DashBoard features, plug-ins, and application details. Details such as software version and ID are also provided.
- **Installation History** tab — This tab displays information on the current and any previous configurations of DashBoard installed on your computer. Details such as the date of installation and applications installed are also provided. From this dialog, you can also Revert to any previously installed software using the Revert button.
- **Plug-ins** tab — This tab displays the ID, name, provider, and version of the currently installed plug-ins, or add-on programs, for DashBoard.
- **Configuration** tab — This tab displays information such as platform details, system properties, and user preferences that can be used for troubleshooting.

#### To view the installation details for your DashBoard client:

1. Launch DashBoard by double-clicking its icon on your desktop.
2. From the main DashBoard toolbar, select **Help** > **About DashBoard** > **Installation Details**. The **DashBoard Installation Details** dialog box opens.
3. To view details on the currently installed software features and options for your DashBoard application:
   - Select the **Installed Software** tab.
   - From the provided list, select the required software feature.
   - Click **Properties**.

4. To view details on your DashBoard installation history:
   - Select the **Installation History** tab.
   - Select the required configuration from the **Previous configurations:** list.
   - The **Configuration contents** pane updates to include a list of the features and options.
5. To view details on the installed plug-ins for your DashBoard application:
   - Select the **Plug-ins** tab.
   - Select the required plug-in from the provided list.
   - Click **Legal Info** to display the plug-in licensing agreement.
   - Click **Show Signing Info** to display a new pane that includes the signing date and the signing certificate for the selected plug-in. Click **Hide Signing Info** to close the pane.
   - Click **Columns** to configure how information is displayed in the **Plug-ins** tab.

## Viewing Error Logs

If you are troubleshooting problems in DashBoard, Ross Technical Support may request that you view the error logs for your local DashBoard application.

**To view the error log for your DashBoard application:**

1. From the main DashBoard toolbar, select **Help** > **About DashBoard** > **Installation Details**. The **DashBoard Installation Details** dialog box opens.
2. Select the **Configuration** tab.
3. Click **View Error Log** to display the error log in the selected application.

# Managing openGear Frames in DashBoard

The DFR-8300 series frames offers remote control and monitoring with the combination of the openGear Network Interface card and the DashBoard Control System™. This allows users to remotely monitor and control parameters on DashBoard Connect compatible devices, such as openGear frames and cards. DashBoard connects to an openGear frame using a TCP/IP LAN connection.

This chapter provides instructions for managing openGear frames in the DashBoard Control System. For details on configuring your openGear frame and Network Controller Card, refer to the ***DFR-8300 Series User Manual*** and the ***MFC-8300 Series User Manual***.

The following topic is discussed:

- Managing openGear Frames in DashBoard

## Managing openGear Frames in DashBoard

When DashBoard is launched, openGear frames are automatically discovered and are available in the Tree View where they can be custom identified, collapsed to view just the frame or opened to view available devices in the frame.

This section includes information for enabling DashBoard to auto-connect to openGear frames, manually adding and re-naming frames to the Tree View, and removing frames from the Tree View.

Refer to the section "**DashBoard Basic Tree View**" on page 4–3 for information on using the Basic Tree View. For information on using the Advanced Tree View, refer to the section "**Using the Advanced Tree View**" on page 4–6. It is assumed that if you have the DashBoard Server and URM feature installed, that your user account is configured for write access to the frames communicating with your DashBoard client.

DashBoard uses the open SLP protocol to locate openGear frames on the network. In larger installations, it is recommended to use an SLP Directory Agent (DA). Contact your IT Department for more information on whether your facility uses an SLP DA.

### Adding openGear Frames to DashBoard

Each openGear frame lists all devices within the frame, and provides status information in the Tree View. You can also remove and disconnect an openGear frame from DashBoard. There are two methods for adding an openGear frame to the tree view: using the auto-connect feature or manually adding a frame by specifying the IP address of the frame. Both methods are described in this section.

#### Using the Automatic Detection Feature

By default, the DashBoard Control System auto-detects any openGear frame on the same IP subnet. How often DashBoard queries the network for new frames (the default is every 10 seconds) depends on how the **Automatic Detection** feature is configured in the **Preferences** menu. Refer to the section "**Automatic Discovery**" on page 7–2 for setup details. Refer to the section "**DashBoard Basic Tree View**" on page 4–3 for information on the toggling the **Auto-Connect Devices** button.

#### Manually adding openGear Frames to DashBoard

You must add openGear frames to the Tree View manually when the frame is on a different subnet from your computer running the DashBoard client.

**To manually add an openGear frame to a DashBoard Tree View:**

1. Click ✚ on the **Tree View** toolbar to open the **Select Equipment or Service Type to Add** dialog box.
2. Expand the openGear node.

3. Select **TCP/IP openGear Frame**.
4. Click **Next >** to open the **New TCP openGear Frame Connection** dialog box. The **Invalid host name or IP address** message remains until you enter text in the **IP Address** field.
5. In the text fields provided, enter the IP Address, display name for the frame, and port of the openGear frame you wish to add.

   You can also click **Detect Frame Information** to enable DashBoard to detect the port, name, unique identifier and other connection information from the specified IP Address. Once DashBoard detects the information, the **Automatically Track Updates to Frame Information** check box is selected so that any changes are automatically updated in DashBoard.
6. Click **Finish** to display the openGear frame in the **Tree View**. Frames added to the Tree View are also displayed in the Advanced Tree View.
7. Repeat the procedure for each openGear frame that you wish to add to the **Tree View**.

## Re-naming an openGear Frame in the Tree View

There are two methods for re-naming an openGear Frame in DashBoard. The first method is for frames manually added to the Tree View as described in the section "**Adding openGear Frames to DashBoard**" on page 3–1. The second method describes how to re-name an auto-detected frame using the DashBoard menu options available on the MFC-8300 series cards. Both methods are described below.

**To re-name a manually added openGear frame:**

1. Right-click the frame you wish to rename.
2. Select **Rename Frame**.
3. Enter the new name for the frame in the text field provided.

**To re-name an Auto-Detected openGear frame:**

1. Right-click the frame you wish to rename.
2. Select **Open** to display a Device Editor tab. Note that the tab title displays the name of the frame, and information about the MFC-8300 series Network Controller Card currently installed in the frame.
3. Select the **Network** tab.
4. Enter a new name for the frame in the **Frame Name** field.
5. Press **Enter**.
6. Click **Apply**.

## Removing openGear Frames from a Tree View

This section outlines how to remove an openGear frame from a Tree View in DashBoard. Once a frame is removed, DashBoard no longer reports the status in the Tree View and you are no longer able to monitor or control the affected devices. If communication with a frame is disconnected via the Disconnect option, the status indicator is light gray until the frame is re-connected. If the status indicator is dark gray, with the rest of the node displaying as normal, a connection cannot currently be established to the device.

**To remove or disconnect an openGear frame from the Tree View:**

1. If the frame you are removing is in a Custom Folder, you must first delete the frame from the Custom Folder before it can be removed from the Tree View.
2. To remove a manually added openGear frame from the Tree View:
   - Right-click the openGear frame you wish to remove.
   - Select ✖ to remove the openGear frame from the Tree View.

3. To disconnect communications to an openGear frame from DashBoard:
   - Toggle to off.
   - Right-click the openGear frame you wish to disconnect.
   - Select .
   - The frame status indicator is grayed out in the Tree View.
4. To re-connect to an openGear frame:
   - Right-click the frame status indicator.
   - Select or toggle to on.

#### Auto-Discovery

Selecting for an auto-detected frame will temporarily remove the frame but the frame will re-appear in the Tree View again due to the auto-discovery feature of DashBoard. You must first disable the auto-discovery feature before you can remove a frame in this instance. Refer to the section "**Automatic Discovery**" on page 7–2 for details on configuring the auto-discovery feature.

However, you can still disconnect from an auto-discovered frame by toggling the **Automatic Discovery** option off.

# Using the DashBoard Interface

DashBoard allows for multiple Device Editor tabs to be active and available on one screen which is useful when a functional path involves more than a single device. These tabs can be saved and recalled as a layout, allowing for quick access to frequently used devices. Layouts can consist of a single device window, multiple device windows displayed full screen in tabs, or multiple devices on a shared screen.

This chapter introduces you to the DashBoard client interface, how to access menus and tabs, and how to manage your layouts.

✶ Contact your I.T. Department if you experience communication issues with DashBoard and are running anti-virus software. You may need to verify that there is an exception in your firewall to allow DashBoard to receive TCP data via Port 5353.

The following topics are discussed:

- DashBoard Interface Overview
- DashBoard Basic Tree View
- Using the Advanced Tree View
- The Device Editor Area
- Using Layouts
- Keyboard Shortcuts
- Using DashBoard Help
- Preferences

## DashBoard Interface Overview

This section includes a brief summary of the DashBoard Control System client interface and its components. **Figure 4.1** displays a DashBoard window that includes the Custom Folders and Layouts View tabs. These tabs are not displayed by default when the DashBoard client is launched for the first time.

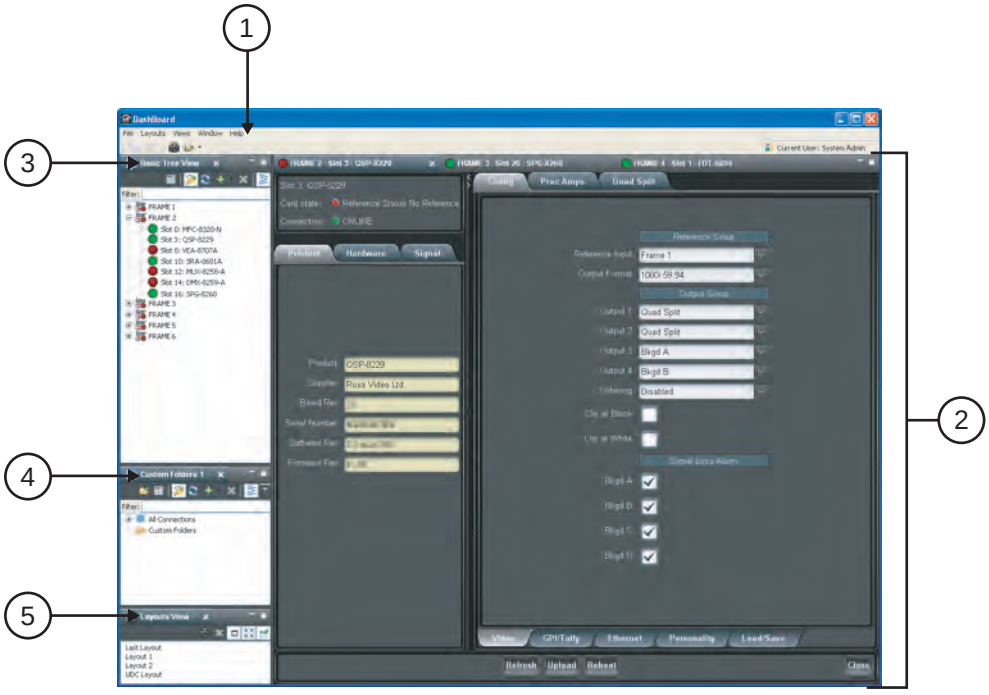



***Figure 4.1*** *DashBoard Interface Overview*

### 1. Main DashBoard Toolbar

The Main DashBoard toolbar provides access to menus that enable you to manually add devices, manage your layouts, and enable different tabs in the DashBoard window.

- **File** — From this menu you can manually add a device (New), save changes made to device configuration files on your computer (Save, Save As..., Save All), log-off from DashBoard when using DashBoard URM (Sign Out), or close the DashBoard client (Exit).
- **Layouts** — From this menu you can display a Layouts View tab in the DashBoard window (Show Layouts View), save your current arrangement of tabs in the DashBoard window (Save Layout), lock the DashBoard window to its current state (Maintain Window State/Size/Location), or select from a list of saved layouts to apply.
- **Views** — From this menu you specify which tabs to display in the current DashBoard window.
- **Window** — From this menu you can open multiple DashBoard windows on a single screen (New Window), restore the default DashBoard client layout of tabs (Refresh Perspectives), show or hide the main toolbar (Show Toolbar), set the window to the maximum screen size (Full-screen), lock the current DashBoard window (Lock Screen), and set preferences for Automatic discovery of devices on your subnet, automatic login, and software updates (Preferences).
- **Help** — From this menu you can access the DashBoard online help system (Help Contents), perform checks for DashBoard software updates (Check for Updates), and view details about your current DashBoard client software (About DashBoard).

### 2. Device Editor

This area displays tabs for each device that you double-click from the Tree View. From this view you can verify the device and connection status, update device parameters, and view read-only device information.When shutting down and then re-starting the DashBoard client, the Device Editor tab state is also saved/restored. Refer to the section "**The Device Editor Area**" on page 4–8 and your device manual for more information.

### 3. Basic Tree View Tab

This area lists the devices, such as openGear frames and the cards installed in each frame, that can communicate with DashBoard. From this tab you can open Device Editors, enable or disable auto connections to devices, re-query the network for new devices, manually add new connections, and delete devices from the Tree View. Refer to the section "**DashBoard Basic Tree View**" on page 4–3 for more information.

### 4. Advanced Tree View Tabs

The Advanced Tree View feature enables you to create a customized hierarchy of folders and subfolders in a single tab, where each folder can be expanded to display a list of devices and/or subfolders. You can re-organize your devices in a Custom Folder tab to suit your workflow by dragging and dropping devices from the Basic Tree View to any open Custom Folder tab. Note that this tab is not displayed by default. For more information on using Custom Folders, refer to the section "**Using the Advanced Tree View**" on page 4–6.

### 5. Layouts View Tab

This feature enables you to save and restore a series of Device Editor tabs and the DashBoard window size and position as a layout. Layouts can be recalled using the options in the Main DashBoard toolbar or from a Layouts View tab. Note that saving/restoring a layout restores the current Device Editor tab selection, the divider position and scroll position in opened Device Editor tabs. Note that this tab is not displayed by default. For more information on layouts, refer to the section "**Using Layouts**" on page 4–10.

## Status Indicators

Some devices include a status indicator beside the node in the Tree Views, custom folders and subfolders. The Frame Status Indicator, the Custom Folders, and the subfolders reflect the most severe status of any contained devices.

Status severity is indicated by color as follows:

- **Green** — This color indicates that the device is running correctly and communicating with the frame.
- **Orange** — This color indicates that the MFC-8300 Network Controller Card for that frame can only support a limited number of connections and that maximum has been reached. You can select the **Force Connect** option, after right-clicking on the frame status indicator, to establish a connection between the frame and your DashBoard workstation. However, doing so will disconnect another connection to the same MFC-8300 Network Controller card.
- **Yellow** — This color indicates a minor problem with the device.
- **Red** — This color indicates that the device has a significant error condition. For example, there is no input or reference signal from the card.
- **Gray** — This color indicates the device is currently offline and cannot communicate with DashBoard. The offline status is also reflected in its **Device Editor** tab.
- icon — If the status indicator is replaced by this symbol, the user does not have permission to view/modify the device. In the case of an openGear frame, this icon means the frame parameters are locked and the Master Password is required to use it. Refer to the ***DashBoard Server and URM User Manual*** and the ***MFC-8300 Series User Manual*** for details. When a device has this icon, there are no editable parameters underneath it.

## DashBoard Basic Tree View

This section outlines the Basic Tree View of the DashBoard client. For details on using the Advanced Tree View feature, refer to the section "**Using the Advanced Tree View**" on page 4–6.

### Overview

The Basic Tree View displays devices, such as openGear frames and cards, in a tree structure. When you launch DashBoard, all devices within the same subnet are auto-detected unless this feature is disabled in the tab toolbar. Refer to the section "**Managing openGear Frames in DashBoard**" on page 3–1 for information on adding openGear frames to the Basic Tree View. For information on connecting your openGear frame using a TCP/IP connection, refer to your frame user manual or your facility IT personnel.

The Basic Tree View also displays the devices and status information of each device, allowing you to monitor and control devices from a single computer. The Basic Tree View includes a Filter feature that enables you to search this hierarchy by entering text into the field. **Figure 4.2** provides an example of a Basic Tree View.

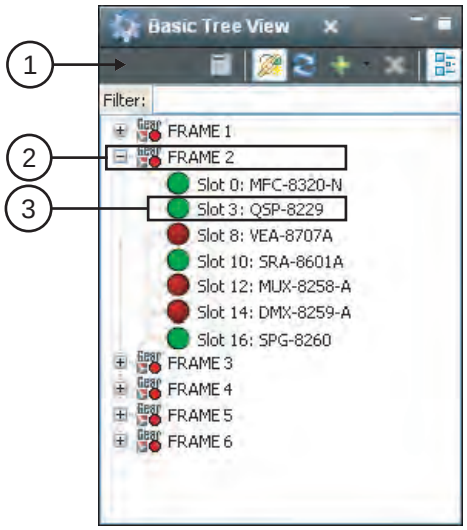



***Figure 4.2*** *Example of a Basic Tree View*

### 1. Basic Tree View Toolbar

This area provides access to the following basic tasks:

- **Device Editor Button** — Selecting this button enables you to view a Device Editor tab of a selected device. To view a Device Editor tab, click the device you wish to edit from the tree view, and select this button. You can also double-click the device from the Basic Tree View list. You can also open additional copies of a Device Editor tab by right-clicking the node and select Open; the active tab is the one displayed in the foreground.
- **Auto-Connect Devices Button** — Toggling this button enables DashBoard to automatically connect to devices and display information in the Basic Tree View. The default setting is enabled (auto-connect).
- **Re-query the Network Button** — Selecting this button enables DashBoard to query the network and automatically add any new devices to the Basic Tree View. Note that DashBoard automatically queries the network approximately every 10 seconds. If Automatic Discovery is disabled, you can force a network query by selecting this button.
- **Add New Connection Button** — Selecting this button opens the **New** dialog box and enables you to manually add a device, such as an openGear frame, to the Basic Tree View. Use this button to add a device that cannot auto-connect but can be found via the network.
- **Delete Button** — Selecting this button enables you to delete a selected offline or manually added device from the Basic Tree View.
- **Group Similar Devices Button** — This button determines how devices are displayed in the Tree View. When toggled on, devices are grouped by class. When toggled off, the Tree View is sorted alphabetically.

### 2. Filter Search Field

Each Custom Folder includes a Filter feature that enables you to search the Tree View by entering text into the field. DashBoard automatically displays the search results in the selected tab under the All Connections node. For example, to search for a UDC-8225, enter 8225 in the **Filter** field and the tab only lists the frames that have a UDC-8225 installed. Expand the frame nodes inside the All Connections node to display the specific slots with UDC-8225 cards. To clear the **Filter** field, delete the text.

### 3. openGear Frame Status Indicator

A status indicator is displayed for each openGear frame detected by DashBoard and is located to the left of the frame name. This status indicator summarizes the current status of the detected devices in that specific openGear frame. For example, FRAME 2 in **Figure 4.2** indicates a red status because at least one card is reporting a red status (in this case there are three). A , or an arrowhead, next to a status indicator signifies that the list can be expanded to display a list of devices installed in that frame.

### 4. Device Status Indicator

A status indicator is listed for each DashBoard Connect compatible device. This icon includes the card status, the slot in which it is installed in that frame, and the device product name. This information is detected automatically. To view a device in the **Device Editor**, double-click its status indicator. Note that if the device is offline, you cannot open a tab for it in the Device Editor area.

## Using the Basic Tree View

This section briefly summarizes how to use the Basic Tree View features.

**To open or close the Basic Tree View tab:**

- To open a tab, select **Views** > **Basic Tree View**.
- To close a tab, right-click the **Basic Tree View** tab, then select **Close**.

**To enable automatic discovery:**

Note that this method is applicable to all tree views.

1. Confirm the Automatic Discovery feature settings by selecting **Window** > **Preferences**. Note that by default, this feature is enabled and DashBoard polls for devices every 10 seconds.
2. Toggle on the **Basic Tree View** toolbar to enable DashBoard to automatically connect to the listed devices in the Basic Tree View.

**To manually add a device to the Basic Tree View:**

1. Click on the **Basic Tree View** toolbar to display the **Select Equipment or Service Type to Add** dialog box.
2. Add a connection by following the on-screen instructions.

**To display a Device Editor tab**

Display a Device Editor tab using one of the following methods:

- From the **Basic Tree View**, double-click the device node.
- From the **Basic Tree View**, select the device to edit. Click on the **Basic Tree View** toolbar.

**To remove a device from the Basic Tree View:**

1. Select the device to remove from the Basic Tree View.
2. Click on the **Basic Tree View** toolbar to display the **Confirm tree item removal** dialog box displays.
3. Click **OK**.

# Using the Advanced Tree View

The Advanced Tree View feature enables you to display a tab that you can customize with a layout of folders. A folder can display any number of devices or subfolders of devices in a single tab. This feature allows you to drag and drop devices into subfolders, enabling you to quickly customize folders as required. For example, you may wish to create a folder that lists only the UDC-8625 cards installed in your facility.

All device information is automatically updated whenever parameters or status changes occur. You can have multiple custom folder tabs open or have multiple Advanced Tree View tabs open in a single DashBoard window.

## Overview

This section summarizes the Advanced Tree View tabs, Custom Folders and subfolders, and the available menu options.

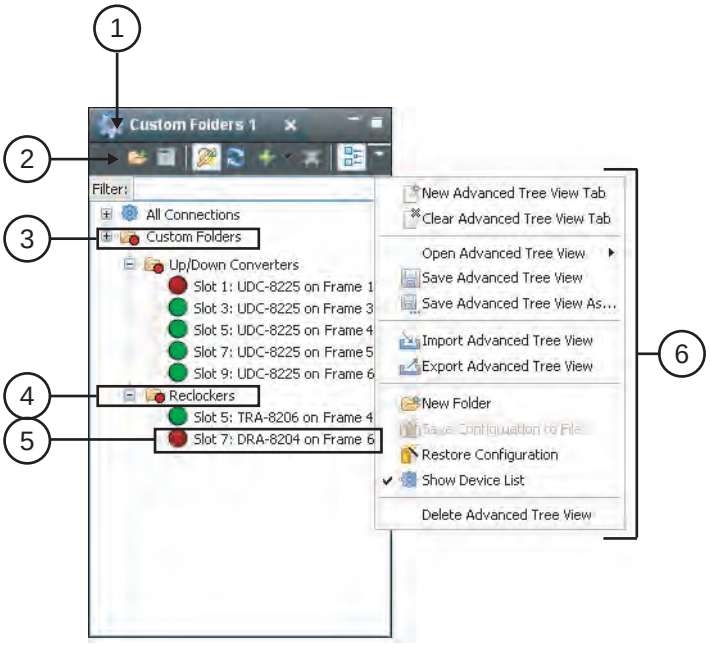



***Figure 4.3*** *Advanced Tree View*

### 1. Custom Folder Toolbar

Like the Basic Tree View, the Custom Folder toolbar includes the Device Editor, Delete and Add New Connection buttons, saving the current tree view, saving and restoring device configuration, and accessing the extra menu options of the Custom Folders tab. In addition, there is a button for creating new subfolders. You can save the current tree view using the button in the main DashBoard toolbar.

### 2. Filter Search Field

Each Custom Folder includes a Filter feature that enables you to search the Tree Views by entering text into the field. DashBoard automatically displays the search results in the selected tab under the All Connections node. For example, to search for a UDC-8625, enter 8625 in the **Filter** field and the tab only lists the frames that have a UDC-8625 installed. Expand the frame nodes inside the All Connections node to display the specific slots with UDC-8625 cards. To clear the **Filter** field, delete the text.

### 3. Custom Folders

Each Advanced Tree View tab includes a main Custom Folder. In this folder, you can create and re-name subfolders to organize devices for customized views. The status indicator represents the current status of the devices in the custom subfolder. If a device in the subfolder needs attention, the status indicator shows the most critical warning level. For example, the Custom Folders icon in **Figure 4.3** indicates a red status because the UDC-8225 in FRAME 1 and the DRA-8204 in FRAME 6 are reporting an error conditions.

Right-clicking a custom folder displays a dialog that includes options for creating a new subfolder, connecting or disconnecting devices, and re-naming the folder. To add devices to a Custom Folder, simply drag a device or frame from the All Connections directory to the desired Custom Folder.

### 4. Custom Subfolders

A Custom Subfolder displays the devices connected to DashBoard that you have specified. Creating Custom Subfolders allows you to group similar devices together that may be installed in different frames, such as the folders in **Figure 4.3**. If a device in the subfolder needs attention, the status indicator shows the most critical warning level. For example, the Reclockers subfolder in **Figure 4.3** indicates a red status because the DRA-8204 in FRAME 6 is reporting an error condition (a red status).

Right-clicking a subfolder displays a dialog that includes options for creating a new sub-subfolder, connecting or disconnecting devices, re-naming the folder, removing the selected subfolder, saving and recalling configuration folders.

### 5. Device Status Indicator

A status indicator is listed for each device in a subfolder. This icon includes the device status, the slot and frame in which it is installed (if it is an openGear card), and the device product name. This information is detected automatically, but you can also re-name an openGear frame or card as required. To view a Device Editor tab, double-click the device node. Right-clicking a device displays a menu that includes options for opening the device, restoring or saving the configuration, re-naming the device or removing the device.

### 6. Custom Folders Extra Options

You may need to select the ▽ button on the **Custom Folder** toolbar to display the available Extra Options.

- **New Advanced Tree View tab** — This option opens a new **Advanced Tree View** tab in DashBoard.
- **Clear This Advanced Tree View tab** — This option closes the current Advanced Tree View and opens a new Advanced Tree View in its place. If you have made any changes to the current Advanced Tree View, you will be prompted to save your work.
- **Open Advanced Tree View** — This option enables you to select a previously saved Advanced Tree View to open in the current session of DashBoard.
- **Save Advanced Tree View** — This option saves the selected Advanced Tree View. If you have re-named the main Custom Folder, the new name is now displayed. An asterisk displays next to the Custom Folder tab name when there are unsaved changes for that tab.
- **Save Advanced Tree View As…** — This option saves the selected Advanced Tree View under a new filename.
- **Import Advanced Tree View** — This option enables you to import an Advanced Tree View from another location or DashBoard computer.
- **Export Advanced Tree View** — This option enables you to export an Advanced Tree View to another location or DashBoard computer.
- **New Folder** — This option enables you to create a new subfolder in the Advanced Tree View.
- **Save Configuration to File** — This option is only available when using the MFC-8310-N and MFC-8320-N Network Controller Cards. For more information on this feature, refer to the section "**DataSafe Overview**" on page 6–1.
- **Restore Configuration** — This option is only available when using the MFC-8310-N and MFC-8320-N Network Controller Cards. For more information on this feature, refer to the section "**DataSafe Overview**" on page 6–1.
- **Show Device List** — Selecting this option displays or hides the list of connected frames and devices in the **Advanced Tree View** tab.
- **Delete Advanced Tree View** — This option deletes the current Custom Folder from DashBoard.

## Using the Advanced Tree View

This section briefly summarizes how to use the Advanced Tree View features.

**To open a new Advanced Tree View tab in DashBoard:**

- From the DashBoard main toolbar, select **Views** > **Advanced Tree View** > **New Advanced Tree View Tab**.

**To add a new subfolder to the Custom Folders:**

1. On the Custom Folders toolbar, select . If you do not see this button, click first to display a list of available functions.
2. You are prompted to enter a name for the new subfolder. An asterisk (*) displays at the top of the Custom Folders tab to remind you to save the this tab.

**To add devices to a subfolder:**

1. Select the device from the **All Connections** or any available Tree View.
2. Drag and drop the device status indicator to the desired subfolder.

   The **Add to Custom Folder?** dialog box displays.
3. Re-name the device, if desired, by entering a new name in the **Name Prefix** field. Note that the new name only applies to the device in the Custom Folder view.
4. Click **OK**.

**To re-name a subfolder:**

1. Right-click the subfolder icon.
2. Select **Rename** to display the **Rename** dialog box.
3. Enter the new name in the **Name:** field.
4. Click **OK**.

**To save an Advanced Tree View tab:**

- On the Custom Folders toolbar, select . If you do not see this button, click first to display a list of available functions.

**To clear an Advanced Tree View tab:**

- On the Custom Folder toolbar, select . If you do not see this button, click first to display a list of available functions.

**To open or close an Advanced Tree View tab:**

- To open a tab, select **Views** > **Advanced Tree View**.
- To close a tab, right-click the **Custom Folders** tab, then select **Close**.

# The Device Editor Area

This section briefly summarizes the Device Editor area. For details on using Device Editor tabs in DashBoard, refer to the chapter "**Configuring Devices**" on page 7–1.

✶ Some devices may not have Device Editors as presented here. Refer to your device manual for specific details on managing your device using DashBoard.

## Overview

The **Device Editor** area displays tabs for devices selected from the tree views. Each device is represented as a tab in the Device Editor area from which you can access the available parameters and menus for that device.You can organize the arrangement of tabs in the Device Editor by dragging and dropping the tabs.

**Figure 4.4** provides an example of a Device Editor tab for an openGear card. In this example, the status indicator, the name of the openGear frame, the slot number that the device is installed in, and the device type are displayed on the top left corner, status information in the bottom left corner, and the available parameters used to control the device in the right-side of the tab. The information and parameters displayed in a Device Editor tab depends on your device and may not reflect what is shown in **Figure 4.4** and described below.

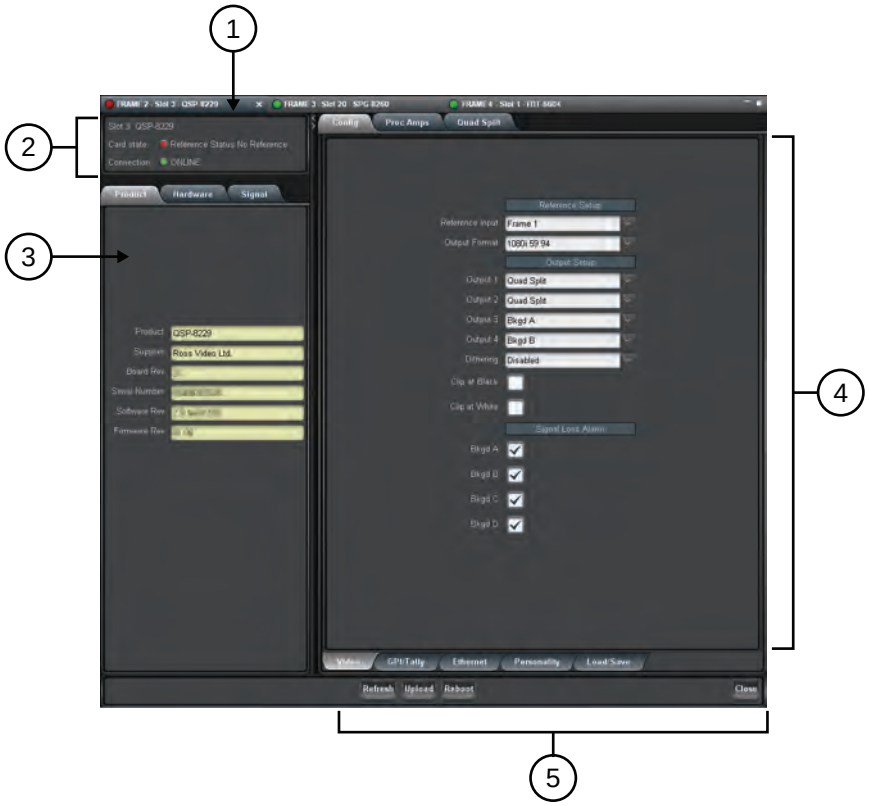



***Figure 4.4*** *Device Tab in Device Editor Area — Example of an openGear Card*

### 1. Device Tab Title

This area displays information to help identify the device such as its status, and the product name. If the device is an openGear card, this tab also displays the openGear frame and the slot that the card is installed in. This information is reported automatically by the device.

### 2. Status Overview

Your device may include an area that reports the operating status and communication activity.

### 3. Read-Only Information

Your device may include an area that displays read-only information such as the status parameters as reported by the device. The parameters and options in this area are dependant on the device selected, but can include the product details such as software versions, hardware information, and signal status. In **Figure 4.4**, the tabs are named **Product**, **Hardware**, and **Signal**, and display as read-only fields.

### 4. Settings and Parameters Area

The contents of this area are dependant on the device selected but can include source selection, video format and timing settings, alarm reporting options, and audio parameters. All changes to openGear card parameters are immediate. Refer to your device manual for details.

**5. Button Area**

The following buttons may be available:

- **Refresh** — Use this button to request the latest information from the device.
- **Upload** — Use this button to upload new software to the device.
- **Reboot** — Use this button to instruct the device to reboot.
- **Close** — Use this button to close the current Device Editor tab.

## Using the Device Editor Feature

This section briefly summaries how to use some of the Device Editor features.

**To display a Device Editor tab:**

Display a Device Editor tab using one of the following methods:

- Double-click a device in a **Tree View**.
- Right-click the device in a **Tree View** and select **Open**.
- Drag and drop the device from the Tree View to the Device Editor area.

**To refresh the parameters of a device:**

- Click **Refresh** on the **Device Editor** tab.

**To maximize or minimize a Device Editor tab:**

1. Right-click the applicable **Device Editor** tab.
2. Select **Maximize** or **Minimize** from the menu.

**To organize the Device Editor tabs:**

Organize tabs using one of the following methods:

- Dock or undock the **Advanced Tree View** and **Layout List** from the DashBoard window by dragging it outside the DashBoard window.
- Drag and drop the Device Editor tabs to organize a layout as required. For information on saving and recalling layouts, refer to the section "**Using Layouts**" on page 4–10.

**To close a Device Editor tab:**

1. Right-click the **Device** tab you wish to close.
2. Select one of the following options:
   - **Close** — Closes the selected Device Editor tab.
   - **Close Others** — Closes all other Device Editor tabs in the group but the highlighted tab.
   - **Close All** — Closes all Device Editor tabs in that Tab Group.

# Using Layouts

This section summarizes the Layout feature of the DashBoard client interface. Information on creating, saving, and managing layouts in the Device Editor is also included.

## Overview

**Layouts** are used to save window configurations in your DashBoard interface. For example, you can save a layout which shows a certain set of devices open in the **Device Editor** area, two Custom Folder tabs, each with a specific size and location on the screen. You can also determine how these **Device** tabs are displayed (tabbed, tiled, or in groups), and whether any Custom Folder tabs are included.

When a layout is saved, it captures details such as how each component of DashBoard is displayed, including the state of any Device Editor tabs, the current tab selection, divider positions between tabs, and scroll position in Device Editor tabs.

Each layout also saves the DashBoard window state, size, and position. Note that the size and position of any undocked tabs are also saved. When restoring layouts from the Layout List, you can toggle whether to use the stored layout, or the current layout, of the open window with the three right-hand buttons on the toolbar.

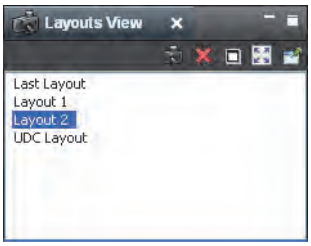



***Figure 4.5*** *Layouts View Tab*

The following buttons are available in the toolbar of the Layout List, from left to right:

- **Save Layout** — Use this button to open the Save Current Layout dialog box.
- **Delete the Selected Layout** — Use this button to delete the currently selected layout in the Layout List.
- **Maintain Window State** — Click this button to keep the DashBoard window at its current state when restoring a layout (maximized or sized). Click the button again to turn off this feature.
- **Maintain Window Size** — Click this button to keep the DashBoard window at its current size when restoring a layout (if not maximized). Click the button again to turn off this feature.
- **Maintain Window Position** — Click this button to keep the DashBoard window at its position on the desktop when restoring a layout. Click the button again to turn off this feature.

## Managing Your Layouts

Once your DashBoard window and **Device** tabs are organized the way you wish, you can save this configuration as a new Layout.

**To display a Layouts View tab:**

Display a Layouts View tab using one of the following methods:

- Select **Layouts** > **Show Layouts View**.
- Select **Views** > **Show Layouts**.

**To organize the tabs:**

1. Click the tab you wish to move.
2. Drag and drop the tab to the new location within the DashBoard window.

**To re-position a tab in DashBoard:**

1. Click the tab you wish to move.
2. Position a tab using one of the following methods:
   - Drag and drop the tab to the new location within the DashBoard window.
   - Undock the tab from the DashBoard window by dragging it outside the DashBoard window.

**To save the current layout:**

1. Click in **Layouts View** tab to display the **Save Current Layout** dialog box.
2. Enter a name for the new layout in the provided text field.
3. Click **OK**.

**To recall a layout:**

Recall a layout using one of the following methods:

- Select **Layouts** from the main DashBoard toolbar. Select a layout from the provided list.
- From the **Layouts View** tab, double-click the name of the layout. DashBoard restores the tab selection, divider position, and scroll position of any Device Editor tabs that were opened when the layout was last saved.

**To rename a layout:**

1. Right-click the layout in the **Layouts View** tab.
2. Select **Rename** to open the **Rename Layout** dialog box.
3. Enter a new name for the layout in the provided text field.
4. Click **OK**. The new name for the layout displays in the **Layouts View** tab.

**To delete a layout:**

1. From the **Layouts View** tab, select the layout you wish to delete.
2. Click to display the **Confirm Layout Delete** dialog box.
3. Click **OK**. The layout is deleted from the **Layout List**.

## Keyboard Shortcuts

The Keyboard Shortcuts feature provides the ability to define a library of commands than can be executed through a series of keystrokes in the DashBoard client, DashBoard Connect compatible products, and other DashBoard product plug-ins. The DashBoard client comes with a standard library, but additional commands may be available depending on the plugs-ins you have installed.

This section provides a brief overview of how to manage your library in the DashBoard client. For information on commands available for your specific DashBoard Connect compatible product, refer to its user documentation.

The following terms are used throughout this section:

- **Keystroke** — A combination of simultaneous button presses on your keyboard. References to keystrokes state the button on your keyboard to press, followed by the "+" symbol, followed by the next button to press. There are single key strokes (e.g. **P**), and keystroke combinations (e.g. **P** + **1**). Buttons such as **Shift**, **Ctrl**, and **Alt** are modifier keys and therefore are not considered as button presses.
- **Command** — An action that is executed whenever the keystroke is performed as defined in the **Keyboard Shortcuts** dialog box.
- **Library** — A collection of commands currently available in your DashBoard client. You can save your library as an *.xml file on your computer and then import this file to other computers running compatible DashBoard clients.

## Overview

From the **Keyboard Shortcuts** dialog box in the DashBoard client, you can change the keystrokes used to trigger commands, delete commands, manage your library, and reset your library to the factory default values.

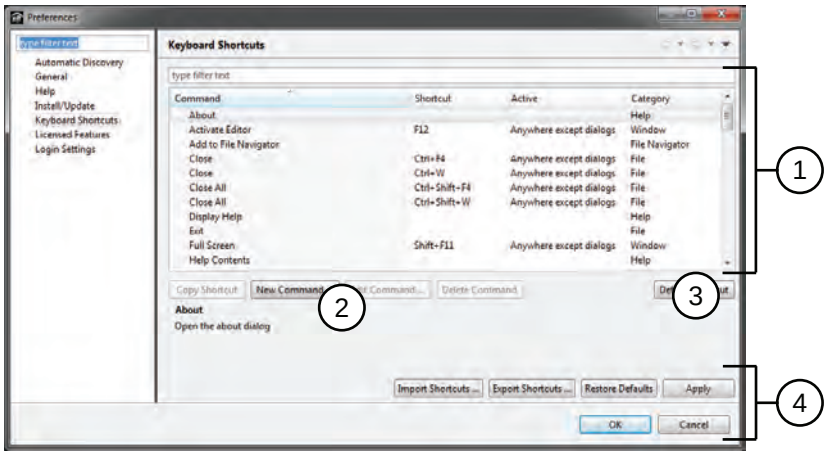



***Figure 4.6*** *KeyBoard Shortcuts Dialog Box*

### 1. Commands Set Area

This area displays a list of the commands currently enabled in the library for your DashBoard client. The commands are listed in a spreadsheet format with the following columns (left to right):

- **Command** — Displays the name of the command.
- **Shortcut** — Displays the keystrokes for the command.
- **Active** — Defines when the command is executed. There are three standard options but additional options may be available depending on the plug-ins in the DashBoard client.
- **Category** — Specifies the application element that the command affects.

### 2. Command Management Area

The following buttons are available in this area:

- **Copy Shortcut** — Creates another instance of the selected command, however the Shortcut (keystrokes) and Active values are not copied.
- **New Command** — Enables you to add a command that is not currently listed in the Commands Set area.
- **Edit Command** — Enables you to change elements of the selected command such as the required keystroke(s), and when to execute the command.
- **Delete Command** — Removes the command from the Commands Set area.

### 3. Default Shortcut Button

Resets only the selected command to its default values.

### 4. Library Management Area

The following buttons are available in this area:

- **Import Shortcuts** — Imports a library to your DashBoard client.
- **Export Shortcuts** — Saves your current library to an *.xml file on your computer.
- **Restore Defaults** — Applies the factory default values to all the commands in your current library.
- **Apply** — Applies your changes made in the Keyboard Shortcuts dialog box. If this button is disabled (grayed out), a conflict is occurring. Refer to the section "**Resolving Conflicts**" on page 4–15 for details.
- **OK** — Applies any changes made to your Keyboard Shortcuts library and exits the Preferences dialog box. If this button is disabled (grayed out), more than one command uses the same keystrokes. Refer to the section "**Resolving Conflicts**" on page 4–15 for details.
- **Cancel** — Exits the dialog box without applying any changes.

## Managing the Keyboard Shortcuts

This section briefly outlines how to perform such tasks as launching the Keyboard Shortcuts dialog box, editing commands, and updating the library.

**To navigate to the Keyboard Shortcuts dialog box:**

1. Select **Window** > **Preferences** to display the **Preferences** dialog box.
2. Click **Keyboard Shortcuts** to display the **Keyboard Shortcuts** dialog box.

**To copy a command:**

1. Select the command from the list in the **Commands Set** area.
2. Click **Copy Shortcut**. The new command is now listed in the Commands Set area.
3. Edit the command as outlined in the section "**To edit a command:**" on page 4–15.

**To edit keystrokes for a command:**

1. Select the command from the list in the **Commands Set** area.
2. Click the appropriate cell in the **Shortcut** column.
3. Type the keystrokes you wish to assign to the command. For example, you could press **D** then press **H** for the Display Help command. The new keystrokes now display in the cell. In the example given, the cell would now display "D, H". If you press and hold the **Ctrl** button and then press **D**, the new keystrokes would display as "Ctrl+D".

**To add a new command:**

1. Click **New Command** to display the **Command Configuration** dialog box.
   - The Command Selection dialog box displays a list of available commands. This list is dependent on the plug-ins you have installed in the DashBoard client.
2. Select a command from the list.
3. In the **Shortcut** field, type the keystrokes you wish to assign to the command.
4. From the **Active** drop-down menu, specify when to execute the command.
5. Click **Next** and follow the on-screen instructions.
6. Click **Finish** to apply your changes, add the new command to the list in the **Commands Set** area, and return to the **Keyboard Shortcuts** dialog box.
7. Click **OK** to apply your changes and exit the dialog box, or click **Apply** to update the library without exiting the dialog box.

**To edit a command:**

1. Select the command from the list in the **Commands Set** area.
2. To edit the keystrokes for the command:
   - Click the appropriate cell in the **Shortcut** column.
   - Type the keystrokes you wish to assign to the command.
3. To specify when to execute the command:
   - Click the appropriate cell in the **Active** column.
   - Select an option from the drop-down menu. The following options come standard:
     - **Anywhere** — Executes the command whenever the DashBoard client is running and active or when a dialog box is active.
     - **Anywhere except dialogs** — Executes the command only when the DashBoard client is active but not when a dialog box is active.
     - **Only in dialogs** — Executes the command only when a dialog box is open and active.
4. Click **OK** to apply your changes and exit the dialog box, or click **Apply** to update the library without exiting the dialog box.

## Resolving Conflicts

A conflict occurs when the same keystroke(s) are assigned to one or more commands. The DashBoard client detects when there are conflicts and flags them in the Commands Set area by displaying a red icon next to the affected command names and setting each name in red. The **Apply** and **OK** buttons are disabled until the conflict is resolved, and a message is displayed near the top of the dialog box. Refer to step 2 in the section "**To edit a command:**" on page 4–15 for details on updating the keystrokes for a command.

Keep the following mind:

- You may elect to keep two or more conflicts active. However, if such commands are triggered, the DashBoard client will prompt you to confirm which command to execute.
- When importing a library, conflicts are also flagged, enabling you to select which command to implement before the library is imported.

## Importing and Exporting Libraries

You can save your library to an *.xml file on your computer and then import that library to another DashBoard client. Note that when importing a library from one DashBoard client to another, ensure that the second client is compatible with the first in terms of DashBoard client software version (DashBoard version 5.1.0 or higher) and installed plug-ins.

**To save your library to a file:**

1. Click **Export Shortcuts** to display the **Export Shortcuts** dialog box. Exporting a library also includes any user-defined commands.
2. Follow the on-screen instructions to save the file to your computer.

**To import a library to your DashBoard client:**

1. Click **Import Shortcuts** to display the **Import Shortcuts** dialog box.
   - The Import Shortcuts dialog box displays the commands currently used in your DashBoard client (Old Shortcut column) and a preview of the commands that will be in your library after the import (New Shortcut column).
   - You can select whether to view only those commands with assigned keystrokes by selecting the **Hide commands with no shortcuts** check box.
   - Importing a library merges commands, including any user defined commands, with the current library.

2. From the **Select shortcuts source:** area, specify a library to import using one of the following methods:
   - Use the drop-down menu to select a file provided by the plugs-ins you have installed.
   - Click to display the **Import shortcuts from file** dialog box and navigate to the file on your computer. Follow the on-screen instructions and click **Open** to return to the **Import Shortcuts** dialog box.
3. Click **OK** to import the library and return to the **Keyboard Shortcuts** dialog box.

### Resetting to Default Values

You can choose to reset a single command or your entire library to the factory default values. Note that resetting the library also deletes any commands that do not come standard with the DashBoard client or your DashBoard Connect compatible product. For example, any commands that are copies will be deleted, and any standard commands you edited will be reset.

**To reset a single command:**

1. Select the command from the list in the **Commands Set** area.
2. Click **Default Shortcut**.

**To reset the library:**

1. Click **Restore Defaults**.
2. Follow the on-screen instructions.

### Restoring Keyboard Shortcut Commands for Copy, Cut, Paste, Undo, and Redo

If you upgraded from DashBoard version 5.1 to DashBoard version 6.0, keyboard shortcut commands for **Copy**, **Cut**, **Paste**, **Und**o, and **Redo** may be inactive.

**To restore these keyboard shortcuts:**

1. In DashBoard, on the **Window** menu, click **Preferences**.

   The **Preferences** dialog appears.
2. In the list on the left side of the **Preferences** dialog, click **Keyboard Shortcuts**.
3. In the **Keyboard Shortcuts** area, in the **Command** list, click **Copy**, and then click **Default Shortcut**.

   Repeat this step for **Cut**, **Paste**, **Undo**, and **Redo**.
4. Click **Apply**.
5. Click **OK**.

## Using DashBoard Help

The DashBoard Help system is accessed by selecting **Help** > **Help Contents** from the main toolbar of DashBoard. The DashBoard Help displays the **Contents** pane and **Search** box in the toolbar by default.

The following Help Systems are currently available in DashBoard:

- **DashBoard Help** — This is the main Help system for the DashBoard Control System that is available by choosing **Help** > **Help Contents** from the main toolbar. DashBoard Help provides information on the various features and options available in DashBoard. Context-sensitive Help is also available by selecting an interface item, such as the Basic Tree View tab, and clicking **F1** (when running Microsoft® Windows®).
- **openGear Help** — This feature provides information on individual openGear devices, such as frames and cards. You can access the openGear Help for your device by selecting the device in the Tree Views and clicking **F1** (when running Microsoft® Windows®) or by choosing **openGear** from the Help dialog box. An option is provided for importing new help files for openGear devices.

- **URM Help** — This feature provides information on the DashBoard User Rights and Management option.

This section briefly summarizes how to configure the DashBoard Help display options and import new help files.

## Configuring the DashBoard Help Display Options

This section summarizes how to configure the DashBoard Help display features:

- **Open Window Context Help** — This is the context-sensitive help that is displayed when you click **F1** (when using Microsoft® Windows®) in the main DashBoard window.
- **Dialog Context Help** — This is the context-sensitive help in a dialog box.

**To configure the Help display options for DashBoard:**

1. Select **Window** > **Preferences** to display the **Preferences** dialog box.
2. Select **Help** to display the **Help** dialog box.
3. Specify the browser used to display the DashBoard Help search results by selecting one of the following options from the **Open help search** menu.
   - **In the dynamic help view** — Select this option to display the DashBoard Help search results in a new pane of the DashBoard interface. This is the default setting.
   - **In a browser** — Select this option to display the DashBoard Help search results in your default web browser.
4. Specify how to display help documents, such as user manuals in *.pdf format, by selecting one of the following options from the **Open help view documents** menu.
   - **In place** — The help document is displayed in the same tab as the link to the document. This is the default setting.
   - **In the editor area** — The help document is displayed in a new tab in DashBoard.
   - **In a browser** — The help document is displayed in your default web browser.
5. Specify how to display the DashBoard Help contents by selecting one of the following options from the **Open help contents** menu.
   - **In the help browser** — The DashBoard Help content is displayed in a new pane of the DashBoard interface. This is the default setting.
   - **In an external browser** — The DashBoard Help content is displayed in your default web browser.
6. Specify how the openGear Help is displayed by selecting one of the following options from the **Open window context help** menu:
   - **In the dynamic help view** — The selected Dialog Context Help is displayed in a new pane of the DashBoard interface. This is the default setting.
   - **In an infopop** — The Help content is displayed as a persistent popup message.
7. Specify how to display the openGear Help by selecting one of the following options from the **Open dialog context help** menu.
   - **In a dialog tray** — The DashBoard Help content is displayed in a new pane of the DashBoard interface. This is the default setting.
   - **In an infopop** — The Help content is displayed as a persistent popup message.
8. Click **Apply** to save your changes. You can also click **Restore Defaults** to disregard any changes you have made.

## Importing openGear Help

DashBoard allows you to download and install additional help files provided by your openGear manufacturer, enabling you to display the most recent help files for your device. There are two methods for importing openGear Help:

- **Importing openGear Help Files** — This method enables you to import a file or directory from a location on your network to your DashBoard workstation. Use this method if you are running Apple® Mac® OS® or Linux® Fedora®.
- **Installing a Ross Video openGear Help Pack** — This method installs an additional DashBoard Help feature that provides a library of Ross Video openGear user manuals. If you are running Microsoft® Windows®, use the procedure provided in the section "**Installing DashBoard Add-on Programs**" on page 2–4. Otherwise, use the procedure for importing openGear Help Files.

✶ The **Include help content from a remote info-center** option is not implemented.

### Importing openGear Help Files

This section briefly summarizes how to import openGear Help Files into DashBoard. If your openGear manufacturer has provided a Help Pack Add-on program and your computer is running Microsoft® Windows®, use the procedure "**Installing DashBoard Add-on Programs**" on page 2–4.

#### To import openGear Help Files:

1. Contact your openGear manufacturer to determine if additional help files are available for your device(s).
2. From the main toolbar, select **File > New** > **Other...** The **Select Equipment or Service Type to Add** dialog box opens.
3. Expand the openGear node.
4. Select **Import openGear Help Files** from the list.
5. Click **Next >**. The **Import openGear Help File(s)** dialog box opens.
6. From the **Add Help From** list, select the type of file to import:
   - **File** — Select this option if the help file is a PDF, or is in a zip file. Proceed to step 7
   - **Directory** — Select this option if the help file has been extracted to a directory. Proceed to step 7
   - **Download** — Select this option if the help file is to be downloaded from the manufacturer website. Proceed to step 8
7. If you selected **File** or **Directory**:
   - Click **Browse...** in the **Selected Help** area.
   - Navigate to the file you wish to import.
   - Click **Open** to return to the **Import openGear Help File(s)** dialog box. Note that the file location is now displayed in the **Selected Help** field.
   - Click **Finish** to import the file into DashBoard.
8. If you selected **Download**:
   - Enter the URL of the help file in the **Selected Help** field.
   - Click **Finish** to import the file into DashBoard.

To view your new help file, choose **Help** > **Help Contents** from the main toolbar and select the file from the **openGear** node.

## Preferences

This section briefly outlines additional options available in the **Preferences** menu in DashBoard.

### Secure Storage

Ross Video recommends that the **Secure Storage** feature be configured only by system administrators, or as directed by Ross Technical Support when troubleshooting.

### Web Browser

The **Web Browser** feature enables you to add, remove, or edit installed web browser applications for your computer running DashBoard. Ross Video recommends leaving the options for this feature at the default states (where the **Use internal Web browser** is selected and the **External Web Browser** is set to the default web browser application for your computer).

✶ The **Web Browser** feature is not yet implemented.

# PanelBuilder™

PanelBuilder is a DashBoard tool for creating custom interfaces for products from Ross Video, openGear partners, and DashBoard Connect Partners. Supported products include openGear cards, DashBoard Connect devices, CamBot robotic camera systems, XPression graphics systems, BlackStorm video servers, NK Routers, and Carbonite and Vision production switchers.

This chapter describes how to create CustomPanel interfaces using PanelBuilder features. It contains the following topics:


- "**About PanelBuilder**" on page 5–1
- "**Creating a CustomPanel**" on page 5–8
- "**Edit Mode**" on page 5–9
- "**Adding Device Editors, Device Summaries, and Device Controls**" on page 5–15
- "**Adding Basic Components**" on page 5–18
- "**Adding Data-Backed Components**" on page 5–34
- "**Timers**" on page 5–40
- "**Assigning Tasks to Buttons, Labels, and Timers**" on page 5–43
- "**Triggering Tasks Externally**" on page 5–50
- "**Editing Components**" on page 5–53
- "**Locking Panel Proportions**" on page 5–86
- "**Parameters and Data Sources**" on page 5–87
- "**Router Control Panels**" on page 5–91


## About PanelBuilder

You can quickly and easily create CustomPanel interfaces tailored to the exact requirements and preferences of each operator. CustomPanels can contain custom controls and labels you create, as well as controls and status indicators imported from any number of DashBoard-enabled devices such as NK Routers and openGear cards.

Within PanelBuilder you can:

- Automate custom workflows by using any or all of the following methods:
  - Drag and drop buttons, indicators, controls, and entire device editors from devices into CustomPanels.
  - Create new buttons, indicators, and controls.
  - Include navigational aids such as tabs, tables, labels, and scrollbars to enhance usability.
  - Customize the appearance of objects by repositioning, resizing, changing colors, specifying fonts, and adding background graphics.
  - Create monitoring and control systems by mixing and matching controls from multiple products on one CustomPanel.

    For example, you can create an interface that includes some or all of your openGear cards. An operator can view the operational status of all devices, and then click a button to modify a given device's configuration settings.
  - Embed web browser windows, enabling users to control devices that provide web-based control interfaces.
- Control CamBot robotic camera systems.

  For example, you can create a CustomPanel that shows the plan view of a studio, with buttons representing each shot you created using the CamBot Control Computer. When the operator taps a button, a CamBot camera system moves into position, ready to capture a perfect shot of the subject.

- Use RossTalk commands to control a variety of Ross Video products including Carbonite and Vision production switchers, and XPression graphics systems.
- Use Video Disk Control Protocol (VDCP) commands to control the Ross Video BlackStorm video server.
- Use custom parameters and imported data to create interfaces that generate and update data for XPression graphics in real-time.

  For example, you can create an interactive sports scoreboard that automatically sends updated scores and player information to XPression for immediate on-air display. The score is updated by the operator. The player information is extracted from an XML file when the operator clicks the player's name. Such a CustomPanel is so easy to use that the operator needs to know only about the sport — not about using DashBoard and PanelBuilder.
- Use Ross Video ogScript to add advanced logic to your interfaces. Most JavaScript objects and functions work in ogScript.
- Use scripting to extend DashBoard's control and monitoring to networked devices that provide an interface for this purpose via UDP, TCP or HTTP.

## CustomPanel Examples

This section describes some CustomPanels created using PanelBuilder.

### Engineering Monitoring and Control Panel

The CustomPanel in **Figure 5.1** includes live controls and indicators within a flow diagram. This example can be used to monitor and control a system of openGear and DashBoard enabled products. Signal presence, audio level meters, source selection, control buttons, and drop down menu selectors from any product in the system can be combined to produce a custom solution that directly addresses your needs.

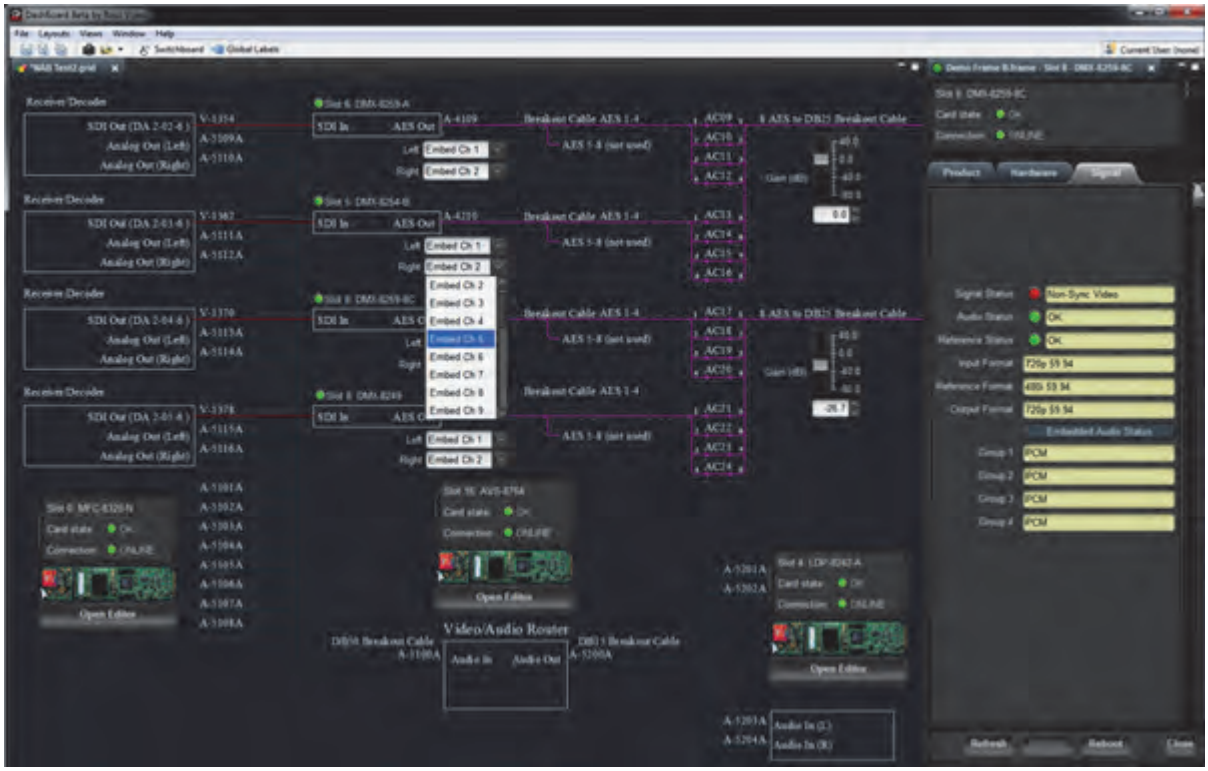



***Figure 5.1*** *- Engineering Monitoring and Control Panel, Created Using PanelBuilder*

### Tennis Scoring Application for XPression

The CustomPanel in **Figure 5.2** is designed to control scoring graphics within a series of XPression Graphics templates designed for tennis. The operator does not need to know anything about DashBoard or XPression - only about scoring tennis.

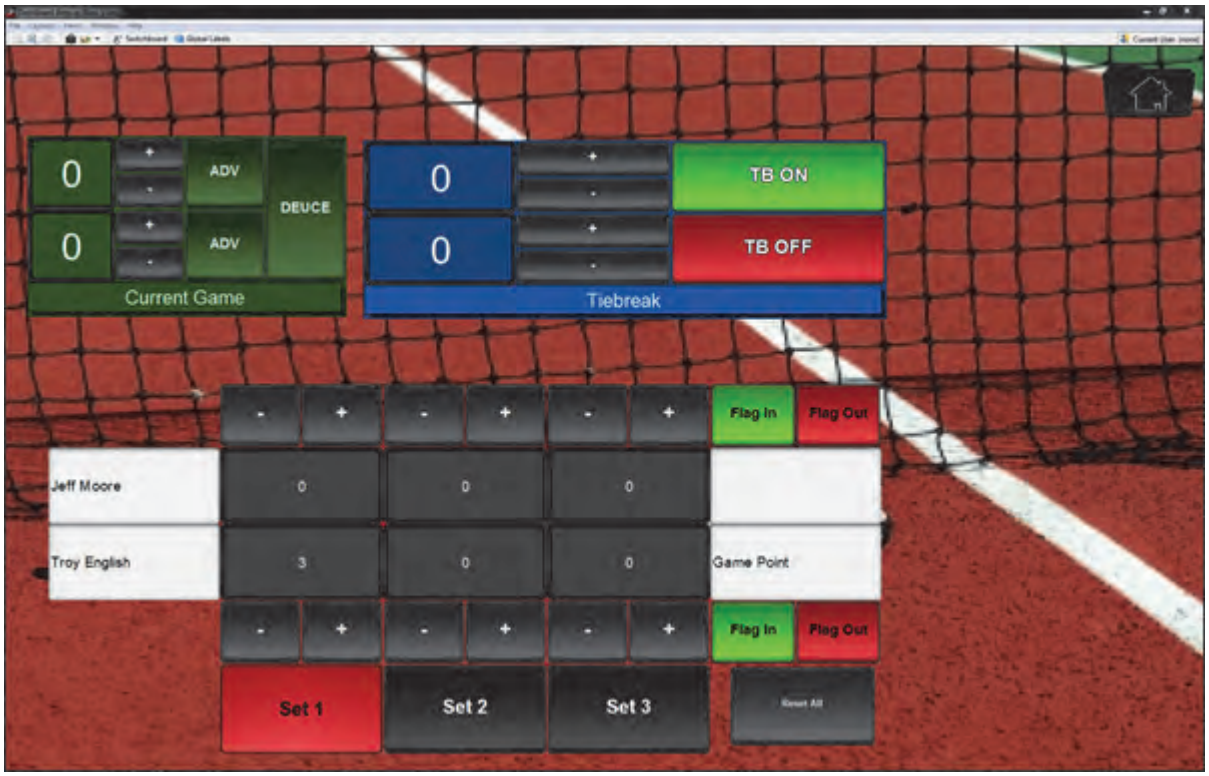



***Figure 5.2*** *- Tennis Scoring Application for XPression, Created Using PanelBuilder*

## Hockey Statistics Application for XPression

The CustomPanel in **Figure 5.3** is designed for use with the Ross XPression Graphics system. The panel pulls in XML statistics about each of the teams and their players. Simply by typing in the jersey number, a player's stats and thumbnail are recalled to the control screen. Buttons in the panel can be used to take templates associated with these stats on and off air.

***Figure 5.3*** *- Hockey Statistics Application for XPression, Created Using PanelBuilder*

### House of Legislature CustomPanel

The Legislative Control System panel in **Figure 5.4** can be adapted to a particular legislative assembly without requiring programming.

This example features buttons representing members. When the operator clicks a member button, a Ross CamBot robotic camera moves to a preset position perfectly focused on the member, the name graphic for the member is prepared, and the camera and shot details are sent to the preview bus of the video switcher. The operator can then press **TAKE** to take the shot directly to air. Any number and layout of seats is possible.

Ross Video offers the Legislative Control System panel as part of a complete automated production system, customized to your legislature and existing delegate systems. For more information, see the **Solutions** tab on the Ross Video website (www.rossvideo.com).

***Figure 5.4*** *- House of Legislature Control Panel, Created Using PanelBuilder*

## PanelBuilder Concepts and Terminology

In PanelBuilder, you create CustomPanels. Each CustomPanel can contain any combination of other PanelBuilder objects. The following table explains common PanelBuilder concepts and terms.

| Concept or Term | Description |
|---|---|
| buttons | Buttons are control components you can add to a CustomPanel, and which the user can manipulate. You can associate tasks with a button so the tasks are performed when the button changes state. You can also specify different visual appearances (styles) for each button state (On, Off, Default). |
| CamBot robotic camera systems | Ross Video CamBot is a line of robotic camera systems which includes robotic heads, lifts, pedestals, and control systems.<br>In PanelBuilder, you can create tasks that send commands to control CamBot robotic camera systems. |
| canvas | A canvas component is an area of a CustomPanel upon which other components can be placed.<br>Canvas types include:<br>• blank canvas — empty area.<br>• image canvas — includes a background image. |

| Concept or Term | Description |
|---|---|
| components | Components are objects on a CustomPanel. Components include tables, labels, tabs, canvases, and buttons. When you drag device controls into a CustomPanel, they become components of the panel. |
| CustomPanel | A CustomPanel is a user interface you create in PanelBuilder. CustomPanels can be used to monitor and/or control a wide variety of Ross Video products. CustomPanels are saved as**.grid** files. CustomPanels are sometimes referred to simply as “panels”. |
| data sources | In PanelBuilder, a data source contains parameter data which can be included and manipulated in a CustomPanel. Device parameter data can be edited to change device settings. You can also create local parameter data in PanelBuilder, and reference local parameters in scripts.<br>Data sources can be XML data files, or parameter data from a device. When you create a CustomPanel, you can opt to automatically create an XML data file to store data for parameters you create in PanelBuilder.<br>You can also create a panel file that includes the parameter data within the CustomPanel. Storing the data in the panel file eliminates the need to for a separate XML file and improves portability.<br>Underlying every CustomPanel is a hierarchy of elements, each of which can be associated with only one data source. If no data source is specified, the element inherits a data source association from its parent element. To view the element hierarchy, enter Edit Mode, double-click an element, and look at the component hierarchy tree in the top left portion of the Edit Component window. |
| device editor | The default monitoring and control interface for a device. When you double-click a device in the Tree View, its device editor opens in the Device View area of the DashBoard window.<br>You can also embed device editors into CustomPanels. |
| device view | The area of the DashBoard window in which CustomPanels and/or device editors are displayed. |
| devices and<br>device controls | For the purposes of this chapter, devices are products that can be monitored and controlled using DashBoard. Devices include NK routers, BlackStorm servers, openGear cards, and DashBoard Connect devices.<br>In PanelBuilder, you can drag openGear device controls from the TreeView or from a device’s editor into CustomPanels. You can also drag the entire device editor into the panel. Device controls can display device status information, or be used to change device configuration settings.<br>You can control BlackStorm servers using VDCP scripts integrated with CustomPanels.<br>You can drag NK router status indicators into CustomPanels. |
| dropspot | The empty area of a new component, upon which you can place parameters or components. For example, when you create a table, each cell of the table is a dropspot. |
| edit mode | PanelBuilder Edit Mode enables you to modify a CustomPanel.<br>When not in edit mode, the panel is ready for use. |
| labels | Labels are short text blocks you can add to a CustomPanel. The text can be static, or based on data. You can associate tasks with a label so the tasks are performed when the label is clicked. |
| OGLML | openGear Markup Language is a set of XML elements and attributes used to define CustomPanels. See also the definition for “**XML source**” on page 5–7. |

| Concept or Term | Description |
|---|---|
| ogScript | ogScript is a programming language developed by Ross Video to interact with DashBoard-enabled devices. It is a subset of JavaScript, with PanelBuilder-specific functions added.<br>In PanelBuilder, you can add advanced functionality and logic to CustomPanels by creating tasks that execute ogScript.<br>For more information about ogScript, see the ***ogScript Reference Guide (8351DR-006)***. |
| parameters | There are two types of parameters:<br>• Device parameters are imported from devices, and can be manipulated to control those devices.<br>• Local parameters are data variables you can create in PanelBuilder. They can be displayed, can be modified by panel users, and can be referenced by scripts.<br>Parameters can be assigned tasks, so that when a panel user changes the parameter value, the task is performed. For example, you can create a parameter to represent text for an XPression graphic. You also create an editable text box associated with the parameter. When a user types text in the box and clicks a button, a task is triggered which sends the text to XPression.<br>Supported data types for parameters include strings, integers, and floats. You can also define a parameter as an array.<br>Parameter data can exist only in certain types of panels:<br>• Self-Contained Data Source Panels<br>• XML Data Source Panels<br>• XPression CustomPanels<br>Panel type is defined when you first create the panel. You can also change it later.<br>For more information about creating and editing parameters, see "**The Add/Edit Parameter Window**" on page 5–87. |
| RossTalk | RossTalk is a communication protocol used to control Ross Video products including XPression graphics systems and Carbonite and Vision production switchers.<br>In PanelBuilder, you can create tasks that send RossTalk commands. Tasks can also include ogScript code that uses the `rosstalk()` object to send RossTalk commands.<br>For more information about the RossTalk protocol, see the document ***RossTalk-Commands (4802DR-403)***. |
| split panels | A split panel is an area that is shared by two canvas-like panels. There is a split bar between the two panels, which users can move to adjust how much of the area is dedicated to each panel.<br>Each panel is a dropspot which can contain parameters and components such as labels, buttons, and canvases. When you create a split panel, you have the option of automatically adding blank canvases to both panels. |
| tables | A table is a grid of canvas-like dropspots (or containers), to which you can add parameters and components such as labels, buttons, and canvases.<br>When you create a table, you have the option of automatically filling it with buttons. |
| tabs | A set of tabs is like a set of stacked canvases. Tabs enable you to re-use an area of a CustomPanel for multiple layers of content. Only one tab of the set is visible at any given time. To switch between tabs, users click a small portion of the tab that protrudes from the stack. This portion of the tab is also called a tab.<br>Each tab is a dropspot which can contain parameters and components such as labels, buttons, and canvases. When you create a set of tabs, you have the option of automatically adding blank canvases to all of them. |

| Concept or Term | Description |
|---|---|
| tasks | Tasks are commands that are associated with control components such as buttons, labels, or timers. When the state of the control component meets specified criteria, the tasks are performed.<br>For example, you can create a button that, when clicked, performs the task of moving a CamBot robotic camera to a pre-determined shot position. |
| timers | Timers are time counters that can be displayed on labels and/or associated with tasks.<br>For example, you can create a timer that triggers a task to be performed every six hours. |
| VDCP | Video Disk Control Protocol is a communication protocol used to control hard disk video servers.<br>In PanelBuilder, you can create tasks that send VDCP commands to BlackStorm video servers. |
| XML source | Each CustomPanel consists of a hierarchical arrangement of components. PanelBuilder uses Extensible Markup Language (XML) code to record all characteristics of the panel. This code is stored in a .grid file.<br>As you create and modify a CustomPanel, the underlying XML source code is automatically updated to record the changes. You can also edit the XML source manually, in the Source tab of the Edit Component window.<br>The set of XML elements and attributes used in DashBoard is called openGear Layout Markup Language (OGLML). |

## Creating a CustomPanel

A CustomPanel is the canvas upon which all other panel components reside.

When you create a CustomPanel, you must select a panel template upon which the panel is based.

For CustomPanels that do not require timers, parameters, or access to device data, the most commonly-used template option is the **Blank Canvas** template.

**To create a CustomPanel:**

1. In DashBoard, select **File** > **New** > **New CustomPanel File**.

   The **Create new CustomPanel File** dialog box appears (**Figure 5.5**).

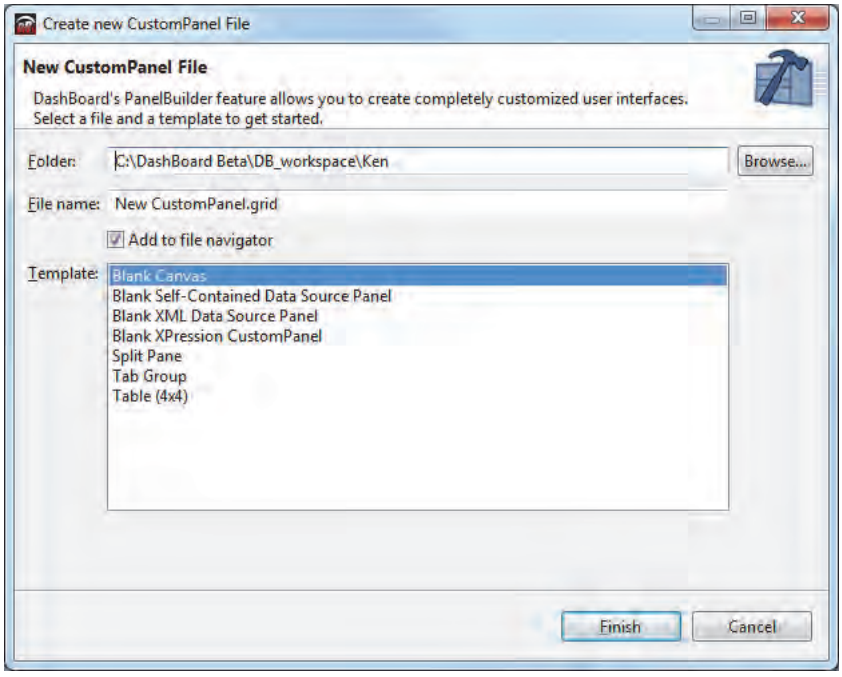



***Figure 5.5*** *- Create new CustomPanel File window*

2. In the **Folder** box, specify where you would like to save the new CustomPanel file.

   **Tip:** If you save it in a directory folder that is listed in the File Navigator, your CustomPanel will also be listed there.

3. In the **File name** box, type a unique name for the new panel file.

4. Use the **Template** list to select a layout for your CustomPanel. Choose from the following:

   - **Blank Canvas** — creates an empty panel without any formatting.

     Use this option if you are creating a monitoring panel that does not require device data, timers, or local parameters.

   - **Blank Self-Contained Data Source Panel** — creates an empty panel file which also stores parameter data.

     Use this option if you are creating a control panel that requires device data, timers, and/or local parameters, and want the parameter data to be stored in the panel file.

   - **Blank XML Data Source Panel** — creates an empty panel file and an associated XML data file for local parameters.

     Use this option if you are creating a control panel that requires device data, timers, and/or local parameters, and want the parameter data to exist in a separate XML file.

   - **Blank XPression CustomPanel** — creates an empty panel and an associated XML data file for parameters.

     Use this option if you are creating a CustomPanel that will export data to an XPression Graphics System.

- **Split Pane** — creates a canvas-like panel that is divided vertically into two equal-sized panes with a split bar between them. The split bar can be moved left and right to adjust the pane sizes. Each pane is a dropspot to which you can add components.
- **Tab Group** — creates a panel with a tab bar at the top of the window. Each tab is a dropspot to which you can add components. The number of tabs can be changed by editing the XML source of the CustomPanel.
- **Table (4x4)** — creates a table consisting of four cells, each of which is a dropspot to which you can add components. The number of rows and columns in the table can be changed by editing the XML source of the CustomPanel.

5. If the folder in which you are saving the panel file is not already listed in the **File Navigator** tab and you want it to be, select the **Add to File Navigator** check box. This enables you to easily and quickly access your CustomPanels in DashBoard.
6. Click **Finish**.

   The new panel appears as a tab in the **Device View**.

   After you create a CustomPanel, you can add device controls and other components using Edit Mode. For more information, see "**Edit Mode**" on page 5–9.

**To open a saved CustomPanel:**

- Do one of the following:
  - › If the CustomPanel (.grid) file is listed in the **File Navigator** tab, double-click it.

    **Tip:** If the **File Navigator** tab is not visible in DashBoard, select **Views** > **File Navigator** in the main DashBoard toolbar.
  - › From the **File** menu, click **Open File**, navigate to the CustomPanel (.grid) file, and click **Open**.
  - › If you are using Microsoft Windows, in **Windows Explorer**, navigate to the CustomPanel (.grid) file, and double-click it.

**To list a CustomPanel in the File Navigator tab:**

1. If the directory folder containing the desired CustomPanel file (*.grid) is not listed in the **File Navigator** tab, select in the **File Navigator** toolbar.

   **Tip:** If the **File Navigator** tab is not visible in DashBoard, select **Views** > **File Navigator** in the main DashBoard toolbar.
2. In the **Browse for Folder** dialog box, navigate to the folder that contains the CustomPanel file, and then click **OK**.

   The folder, including its .grid files, appears listed in the **File Navigator** tab.

## Edit Mode

Once a CustomPanel is created, you can add device controls and components using Edit Mode. Edit Mode is not activated when you first create or open a CustomPanel.

**To enter or exit Edit Mode:**

- In the top toolbar, click **PanelBuilder Edit Mode**.

  Alternatively, you can press **CTRL+G**.

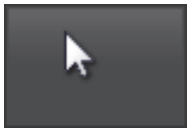
## The Edit Mode Toolbar

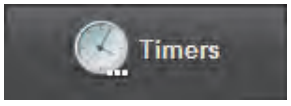
The Edit Mode toolbar (**Figure 5.6**) consists of buttons that enable you to create, resize, reposition, and configure components, timers, and parameters. The selection of buttons varies depending on whether a data source is associated with the CustomPanel. There are also special buttons for panels that contain controls for NK routers.

***Figure 5.6*** *- Three Versions of the Edit Mode Toolbar (without associated data source, with associated data source, and including router control buttons)*

**Table 5.1** describes the buttons that are always available in the edit mode toolbar.

***Table 5.1 - Buttons that are Always Available on the Edit Mode Toolbar***

| Button | Description |
|---|---|
|  | **Click on components** — This pointer mode enables you to manipulate components as though PanelBuilder was not in Edit Mode.<br>For example, in **Click Component** mode you can push buttons or switch between tabs in a tab group. |
| | **Select/Drag Components** — This pointer mode enables you to select components while in Edit Mode. Double-click a component to open the Edit Component window.<br>**Tip:** To select a tab, click the tab panel, and then **CTRL+click** the tab you want to select. |
| | **Move Components** — This pointer mode enables you to reposition components.<br>Drag components to reposition them. Repositioning does not move a component in or out of a containing element such as a tab or canvas. To move a component from one container to another, cut and paste the component. |
| | **Resize Components** — This pointer mode enables you to resize components.<br>Drag the corners of components to resize them. |
| | **Tab and Split** — Reveals toolbar buttons used to create tab components and split panels. |
| | **Tab** — Inserts a tab component.<br>Tabs enable you to include make efficient use of limited screen space.<br>For more information, see "**Tab Groups**" on page 5–21. |
| | **Split Pane** — Inserts a split panel.<br>A split panel consists of two canvas-like areas (panels) with a split bar between them. Each panel is a dropspot that can contain components. Users can move the split bar to adjust how much of each panel is shown.<br>For more information, see "**Split Panels**" on page 5–23. |
| | **Basic Canvas** — Inserts a blank canvas region.<br>Canvases are used to group components. Items placed on a canvas move with the canvas.<br>For more information, see "**Blank Canvases**" on page 5–20. |
| | **Image** — Inserts a canvas that has a background image.<br>For more information, see "**Image Canvases**" on page 5–22. |
| | **Grids / Tables** — Reveals toolbar buttons used to create tables, simple grids, and flow panels. |

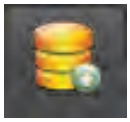
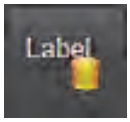
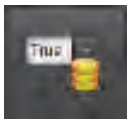
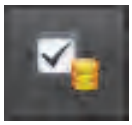
| Button | Description |
|---|---|
| | **Table** — Inserts a table with a specified number of rows and columns.<br>Use tables to align and position multiple rows and columns of components on a single CustomPanel.<br>For more information, see "**Tables**" on page 5–25. |
| | **Simple Grid** — Inserts a simple grid, which is like a table in which each cell is the same size.<br>For more information, see "**Simple Grids**" on page 5–26. |
| | **Wrap Content** — Inserts a flow container, which is like a table. As you add components to a flow container, each is added to the right of the previous one. When a row is filled, additional components appear in the next row.<br>For more information, see "**Flow Containers (Wrap Content)**" on page 5–27. |
| | **Label** — Inserts a text field, known as a label.<br>Labels are read-only for users and can be used to identify components or provide information. Labels can also be associated with a list of tasks which are performed when the user clicks the label.<br>For more information, see "**Labels**" on page 5–28. |
| | **Button** — Inserts a button.<br>Buttons can also be associated with a list of tasks which are performed when the user clicks the button.<br>For more information, see "**Buttons**" on page 5–30. |
| | **Lines** — Inserts a line segment.<br>Line segments can be single lines, compound lines with 90degree bends, diagonal lines, or closed shapes.<br>For more information, see "**Line Segments**" on page 5–32. |
| | **Panel Link** — Inserts a link to another DashBoard panel, such as a device editor or another CustomPanel.<br>Device editors enable you to view and modify configuration settings for the device.<br>For more information, see "**Links to Device Editors or Other CustomPanels**" on page 5–29. |
| | **Browser** — Displays a web page, based on a URL you provide.<br>Web pages are fully functional, but do not include typical browser features such as an address bar or forward and back buttons.<br>For more information, see "**Web Browser Instances**" on page 5–34. |
|  | **Timers** — Allows you to add or edit timers.<br>Timer data can be displayed on labels. Timers can also trigger tasks.<br>For more information, see "**Timers**" on page 5–40. |

If the CustomPanel is associated with one or more data sources, the Edit Mode toolbar includes additional buttons for creating objects that display and/or manipulate data parameters. Data sources include devices from which controls have been imported, and XML data files that have been associated with the panel.

If your panel uses a device as a data source, but is not associated with an XML data file, the Edit Mode toolbar includes only one additional button: **Data Sources**.

**Table 5.2** describes buttons for creating components associated with a data source.

***Table 5.2 - Buttons for Creating Components Associated with a Data Source***

| Button | Description |
|---|---|
| | **Param** — Reveals toolbar buttons used to create components related to data sources and parameters. |
|  | **Display or edit a parameter backed by your data source** — Displays a parameter on the CustomPanel.<br>Drag to define the area, and then specify the parameter to display. You can also allow users to change the parameter value.<br>For more information, see "**Parameter Displays**" on page 5–35. |
|  | **Label** — Inserts a label showing data from a parameter.<br>You can select an existing parameter, or create a new one. Data-backed labels are read-only to the user.<br>For more information, see "**Data-Backed Labels**" on page 5–36. |
| | **Insert an editable text area backed by your data source** — Inserts an editable text field.<br>You can format the text field using the options provided by the data source library. The entered data is stored in the associated parameter.<br>For more information, see "**Editable Text Areas**" on page 5–36. |
|  | **Insert a choice (list, toggle buttons, or radio buttons) backed by your data source** — Inserts a list, toggle, or radio button group on your CustomPanel.<br>The selected data is associated with a parameter from the data source.<br>For more information, see "**Option Choice Controls**" on page 5–37. |
| | **Insert a number (slider, counter, etc.) backed by your data source** — Inserts a numerical entry component.<br>Various formats are available (text entry, sliders, faders, etc.) that allow you to customize how the user enters numeric data. The entered data is stored in the associated parameter.<br>For more information, see "**Numeric Choice Controls**" on page 5–37. |
|  | **Insert a Toggle Choice (checkbox or single toggle button) backed by your data source** — Inserts a toggle component.<br>This component requires the user to make a choice between two states. Choose between check boxes and toggle switches to customize how the user selects a state. The selected state data is stored in the associated parameter.<br>For more information, see "**Toggle Choice Controls**" on page 5–38. |

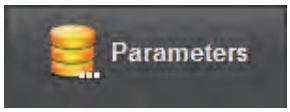
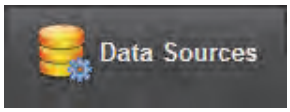
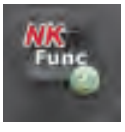
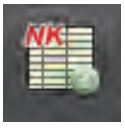
| Button | Description |
|---|---|
|  | **Add or modify data parameters in your data source** — Opens the Add/Edit Parameters window, which lists parameters from your XML data source, and enables you to edit their properties.<br>Parameters are local variables. You can modify them programmatically, or allow users to modify them.<br>For more information, see "**Parameters and Data Sources**" on page 5–87. |
|  | **Modify data sources for your CustomPanel** — Opens the Data Sources window, which lists data sources associated with the CustomPanel, and enables you to edit them.<br>For example, you can create a CustomPanel based on a particular production switcher, then re-associate the panel with another switcher.<br>For more information, see "**Parameters and Data Sources**" on page 5–87. |

**Table 5.3** describes buttons for creating NK Router control components.

***Table 5.3 - Edit Mode Toolbar Buttons for Adding Router Controls***

| Button | Description |
|---|---|
| | **NK NK** — Reveals toolbar buttons used to create components related to router control.<br>For more information, see "**Router Control Panels**" on page 5–91. |
| | **Insert a list of IPS selectors** — An IPS selector list enables you to choose a set of routers to control. |
| | **Insert a list of destinations** — Destinations are video router outputs. |
| | **Insert a list of sources** — Sources are video router inputs. |
| | **Insert a list of levels** — Levels ensure that a certain set of inputs can only be routed to a certain set of outputs. |
|  | **Insert a function button (chop, take, configure, etc)** — Functions are router commands. |
|  | **Insert a level status table** — The level status table lists levels and the sources and destinations associated with them. |

## Adding Device Editors, Device Summaries, and Device Controls

Devices are Ross Video products that can be monitored and controlled using DashBoard. Devices include NK routers, openGear cards, and DashBoard Connect devices.

When you add a device editor, device summary, or device control to a CustomPanel, the device acts as a data source for the panel. Other panel components can access parameter data from the device.

### Device Editors and Device Summaries

A device editor is the default monitoring and control interface for a device. It is generated based on layout data and parameter data provided by the device. You can embed device editors into CustomPanels. The layout of an embedded device editor cannot be edited.

Device summaries are displays of device status information, and can be shown as status dots (with or without label) or full summaries (**Figure 5.7**). You can create a customized device monitoring interface by dragging device summaries from the DashBoard Tree View (or Advanced Tree View) into a CustomPanel.

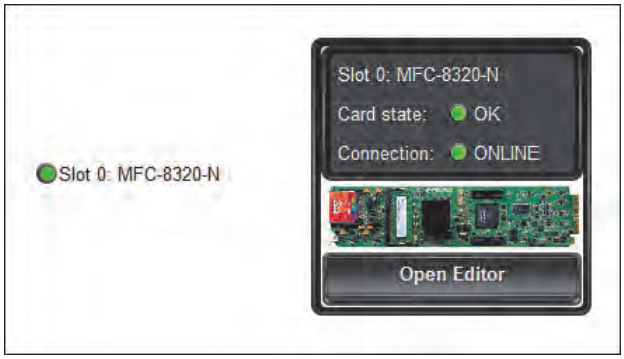



***Figure 5.7** - Two Ways to Display a Device Summary; as a Status Dot with Label (left), and as a Full Device Summary*

If a user hovers over a status dot, a tooltip shows the status as text. If they click the status dot for a frame, a list of status information for the frame and all its cards appears. If they click a card in the list, the device editor for that card opens.

**To add a device editor or device summary:**

1. With the CustomPanel open, enter **PanelBuilder Edit Mode**.

   **Tip:** To enter Edit Mode, press **CTRL+G**.

2. In the DashBoard Tree View, locate the device.
3. Drag and drop the device from the Tree View onto the CustomPanel.

   The **Insert into Component** dialog box appears.

4. Select one of the following:

   - **Device Summary** — shows detailed device summary information, and may include a link for opening the device's editor. The link appears only if the device has an editor.
   - **Embedded Editor —** shows the default device monitoring and control interface for the device.
   - **Status Dot** — shows a color-coded dot to indicate device status.

5. If you want to include the device's label, select **Show Label**.

   **Tip:** Alternatively, you can create a custom label later instead of importing the device's label.

6. Click **OK**.

   The device editor or device summary appears on the CustomPanel.

## Device Controls

Device controls are interface elements that enable users to view configuration settings and to configure device properties. You can create a device control panel by opening a device's editor and dragging individual device controls or groups of device controls into CustomPanels **Figure 5.7**.

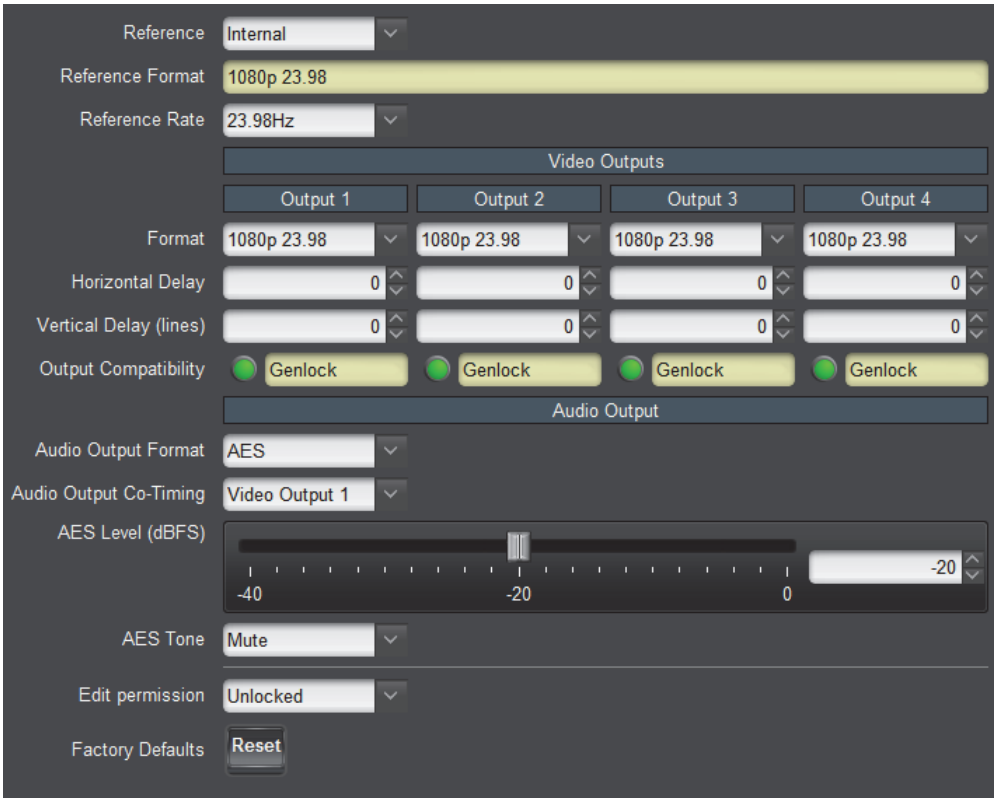



***Figure 5.8*** *- Example of a Group of Device Controls*

**To add device controls:**

1. In the **DashBoard Tree View**, double-click the device to open its **Device View**.
2. In the **Device View**, enter **Edit Mode**.

   **Tip:** To enter Edit Mode, press **CTRL+G**.
3. Rearrange the **CustomPanel** view and **Device View** to make both visible.

   **Tip:** Drag the tab of a view to undock it. As you drag it around the screen, a red rectangle appears, to indicate where the view will be shown after you drop it.
4. Click and drag the control(s) you want to add from the **Device View**, and drop them onto the CustomPanel.

   **Tips:**

   - You can select a single control or a group of controls. As you hover the Device View, the selected control(s) are indicated by a white outline (**Figure 5.9**). When you click to select them, the outline turns bright green.
   - If you click a single control and then press the cursor up button, the set of controls to which the single control belongs is selected.
   - When you drop controls onto the CustomPanel, be sure that the correct destination (container element) is outlined in white.

***Figure 5.9*** *- A Device View, with a Group of Device Controls Selected (bright green outline)*

After you drop the control(s), the **Insert into Component** dialog box appears. The options shown depend on whether you are adding a single control or multiple controls, and on whether or not you are dropping them into a container element such as a canvas or table cell.

5. If you added only a single control, in the **Parameter View Options** area, select one of the following:

   - **Include Parameter Name** — The name of the control, as defined on the device, is also displayed. Alternatively, you can create a custom label in PanelBuilder.
   - **Keep returned elements together** — Applies to parameters that return multiple controls, such as a set of buttons.

     When selected, returned elements can only be modified as a group, and are displayed together neatly. For example, if placed on an absolute position canvas, they do not overlap.

     When not selected, returned elements can be individually modified. For example, you can apply different style options to each element, or position them in separate table cells.

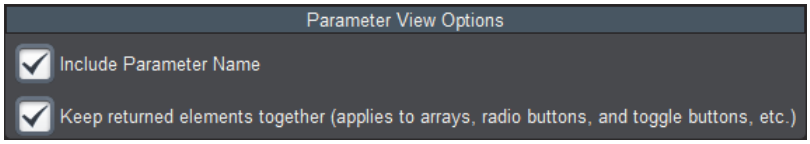

Parameter View Options
Include Parameter Name
Keep returned elements together (applies to arrays, radio buttons, and toggle buttons, etc.)

***Figure 5.10*** *- Parameter View Options*

6. If you added a group of controls, in the **Menu Import Options** area, select one of the following:

   - **Link to menu** — Formatting of controls is inherited from the device, and is not editable.
   - **Import menu** — Formatting of controls is editable, but if they change on the device, the changes are not reflected on the CustomPanel.

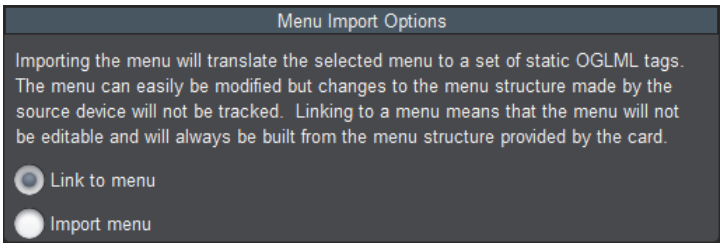

Menu Import Options
Importing the menu will translate the selected menu to a set of static OGLML tags. The menu can easily be modified but changes to the menu structure made by the source device will not be tracked. Linking to a menu means that the menu will not be editable and will always be built from the menu structure provided by the card.
Link to menu
Import menu

***Figure 5.11*** *- Menu Import Options*

7. If you dropped the control(s) onto a container element, in the **Add Data Source/Device Control to CustomPanel** area, specify the scope of elements with which the device's data source is to be associated:

   a. In the right-most drop-down list, select the container element with which to associate the data source.

   b. In the left-most drop-down list, select one of the following options:

   - **Add to** — associates the data source with the selected container element.

     This is the default if the selected container does not already have an associated data source. This option makes it easy to add additional controls from the same device to this container.

   - **Insert Before** — creates a container for the data source above the selected container element.

     This is the default if the selected container is already associated with a data source.

   - **Insert After** — creates a container for the data source beneath the selected container element.

   **Note:** Each element in the component hierarchy can be associated with only one data source. Components that do not have an associated data source inherit a data source association from their parent element.

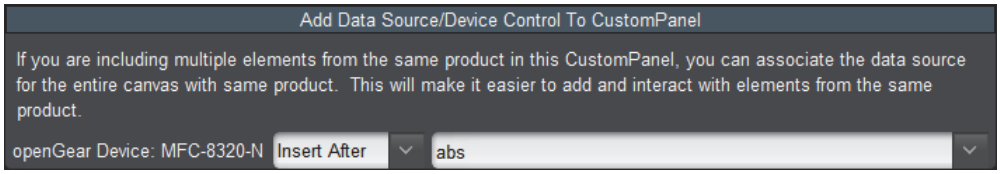

Add Data Source/Device Control To CustomPanel
If you are including multiple elements from the same product in this CustomPanel, you can associate the data source for the entire canvas with same product. This will make it easier to add and interact with elements from the same product.
openGear Device: MFC-8320-N
Insert After
abs

***Figure 5.12*** *- Add Data Source/Device Control To CustomPanel*

8. Click **OK**.

   The device controls are added to the CustomPanel as new components.

## Adding Basic Components

This section includes generic steps for creating basic components such as blank canvases, tabs, split panels, image canvases, tables, labels, buttons, line segments, web pages, and links to devices and to other CustomPanels. These basic components do not use data sources.

**For More Information on...**

- For information about data-backed objects, see "**Adding Data-Backed Components**" on page 5–34

**To add a component:**

1. With the **CustomPanel** open, enter **PanelBuilder Edit Mode**.

   **Tip:** To enter Edit Mode, press **CTRL+G**.

2. On the **Edit Mode** toolbar, click the button corresponding to the type of component you want to add.

   For a list of Edit Menu buttons, see "**The Edit Mode Toolbar**" on page 5–10.

3. On the **CustomPanel**, click and drag to specify the area of the new component (**Figure 5.13**).

   **Note:** Do not drag Edit Menu buttons onto the panel. Click and hold where you want one corner of the component to appear, and drag to define the shape and size of the new component.

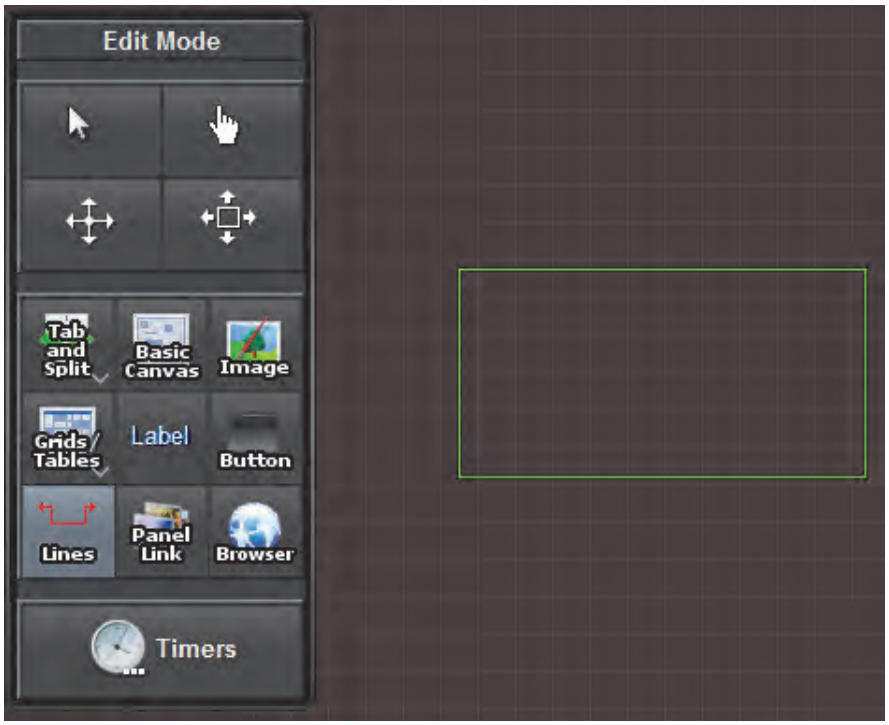



***Figure 5.13*** *- Adding a Component (with **Lines** button selected, dragging to define line boundaries)*

The **Insert into Component** dialog appears.

4. Specify the properties as required.

   Which properties appear in the **Insert into Component** dialog box depends on the type of component you are adding.

5. Click **OK**.

   The component appears on the CustomPanel.

**For More Information on...**

- For detailed information about each type of component, see the corresponding section:
  - "**Blank Canvases**" on page 5–20
  - "**Tab Groups**" on page 5–21
  - "**Split Panels**" on page 5–23
  - "**Image Canvases**" on page 5–22
  - "**Tables**" on page 5–25
  - "**Simple Grids**" on page 5–26
  - "**Flow Containers (Wrap Content)**" on page 5–27
  - "**Labels**" on page 5–28
  - "**Links to Device Editors or Other CustomPanels**" on page 5–29
  - "**Buttons**" on page 5–30
  - "**Line Segments**" on page 5–32
  - "**Web Browser Instances**" on page 5–34
- For information about customizing components by editing their attributes, see "**Editing Components**" on page 5–53.

## Blank Canvases

A canvas is an area that can contain other components, including other canvases. Canvases are used to group components. When you reposition a canvas, its components move with it.

**To create a blank canvas area:**

1. On the **Edit Mode** toolbar, click the **Basic Canvas** button.
2. Drag a box on the panel to define the canvas area. The area outlined in green in **Figure 5.14** is a blank canvas area.

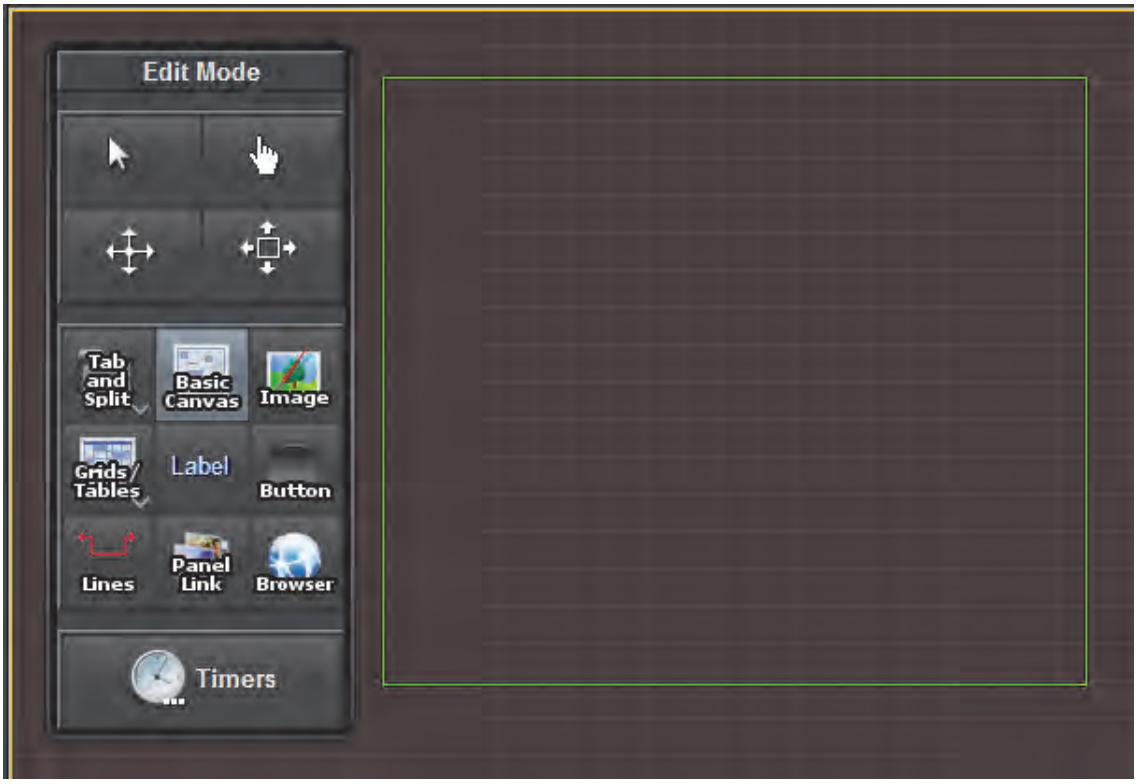



***Figure 5.14*** *- Adding a Blank Canvas (with* ***Basic canvas*** *button selected, dragging to define canvas area)*

> **Tip:** After you create a canvas, by default it does not include a border or fill, so it may be difficult to find in the CustomPanel. To locate the canvas, hover the mouse pointer over the area until a white border appears around the canvas area.

### Editing Blank Canvas Attributes

After you create a blank canvas, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For blank canvases, the Edit Component window contains the following tabs:

- **Abs Attributes Tab** — For more information, see "**Abs Attributes Tab**" on page 5–58.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Tab Groups

A tab group contains one or more tabs, which can contain other components.

**Tips about Tabs:**

- **Tab placement** — When you create a tab group, you specify what side of the tab group the tabs appear on.

  If you select the center option, the tab group does not include any tabs. It is like a set of stacked canvases. You can use scripting to switch which tab is on top. This allows you to use a single area for many purposes.
- **Selecting a tab to edit** — To select a tab in **Edit Mode**, press **CTRL** and click the tab you want to edit.

**To create a tab group:**

1. On the **Edit Mode** toolbar, click the **Tab** button.

   **Tip:** If the **Tab** button is not visible, click the **Tab and Split** button to reveal the **Tab** button.
2. Drag a box on the panel to define the tab group area.

   The **Insert into Component** dialog appears.
3. Specify the tab placement and number of tabs.
4. If you want each tab to contain a blank canvas, select the **Create blank canvases in tabs** option.

   If you do not select this option, when you later add a component to a tab, the component resizes to occupy the entire tab area.
5. Click **OK**.

   **Tip:** By default, the tabs are located along the top edge of the canvas.

   The tab group appears. **Figure 5.15** shows a tab group with three tabs.

***Figure 5.15*** *- An empty tab group*

### Editing Tab Group Attributes

After you create a tab group, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For tab groups, the Edit Component window contains the following editing tabs:

- **Tab Attributes Tab** — For more information, see "**Tab Attributes Tab**" on page 5–77.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Image Canvases

An image canvas displays an image behind all the other components on the canvas. The image can be a .jpg, .png, or .gif file.

**To create an image canvas:**

1. On the **Edit Mode** toolbar, click the **Image** button.
2. Drag a box on the panel to define the canvas area.

   The **Insert into Component** dialog appears.
3. From the **Background Color** list, select a color to fill the area with the specified color. Select [no color] if you do not want to apply a fill.
4. In the **Background (URL)** box, specify the file you want to display in the canvas area.
5. From the **Background Fill** list, select one of the following options to specify how the image is formatted:
   - **Crop** — Fills the entire area with the image while maintaining the aspect ratio. Crops areas that do not fit.
   - **None** — Does not resize the image in any way.
   - **Horizontal** — Stretches the image to fit the horizontal space.
   - **Vertical** — Stretches the image to fit the vertical space.
   - **Shrink** — If the image is too large to fit, scales the image. If the image is too small, does not resize.
   - **Fit** — Scales the image to fit, while maintaining the aspect ratio.
   - **Tile** — Repeats the image as a series of tiles to fill the space.
   - **Both** — Stretches the image horizontally and vertically to fill the space.
   - **Paint9** — Divides the image into nine areas (defined with Background Insets) to define fixed corners, vertically or horizontally stretched sides, and a stretched center.
6. From the **Background Alignment** list, select and alignment option to specify how the image appears within the canvas area.
7. Click **OK** to apply the changes.

   The image canvas appears. **Figure 5.16** shows an image canvas with an image of a production switcher.

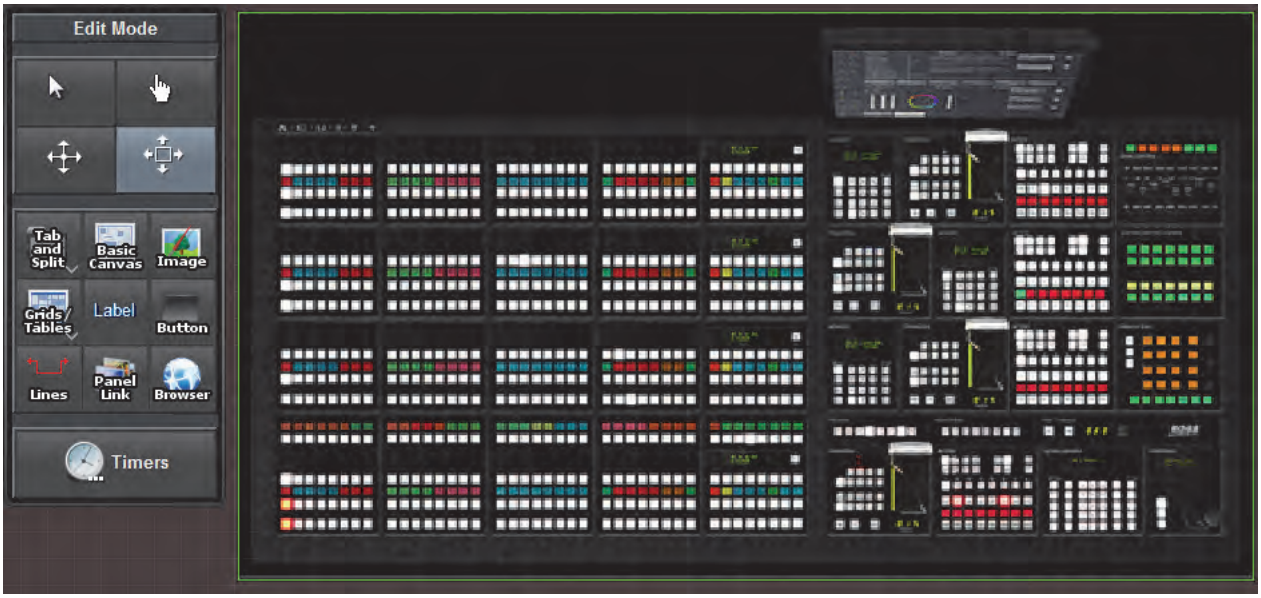



***Figure 5.16*** *- Adding an Image Canvas, showing a picture of a Ross Video Acuity production switcher*

### Editing Image Canvas Attributes

After you create an image canvas, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For image canvases, the Edit Component window contains the following tabs:

- **Abs Attributes Tab** — For more information, see "**Abs Attributes Tab**" on page 5–58.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Split Panels

A split panel has vertical and/or horizontal split bars which partition it into smaller panels. Each panel can in turn be split, and can contain other components. The user can move the split bars to adjust how much of the split panel area is dedicated to each panel.

**Tips about Split Panels:**

- **Selecting all or part of a split panel** — As you hover over a split panel, a border appears, indicating which component is selected. The component can be the entire split panel area, an individual panel, or both parts of a split panel. To select both parts of a split panel, hover over the bar between the two panels.

**To create a split panel:**

1. On the **Edit Mode** toolbar, click the **Split Pane** button.

   **Tip:** If the **Split Pane** button is not visible, click the **Tab and Split** button to reveal the **Split Pane** button.

2. Drag a box on the panel to define the split panel area.

   The **Insert into Component** dialog appears.

3. In the **Orientation** list, specify whether you want split the panel horizontally or vertically.

4. Drag the **Division** slider to specify the default position of the split bar.

   **Tip:** For vertically-split panels, lower values position the bar toward the top. For horizontally-split panels, lower values position the bar toward the left.

5. If you want both parts of the split panel to contain a blank canvas, select the **Create blank canvases** option.

   If you do not select this option, when you later add a component to one of the two panels, the component resizes to occupy the entire panel.

6. Click **OK**.

   The split panel appears. **Figure 5.17** shows a split panel, with different background colors for each pane.

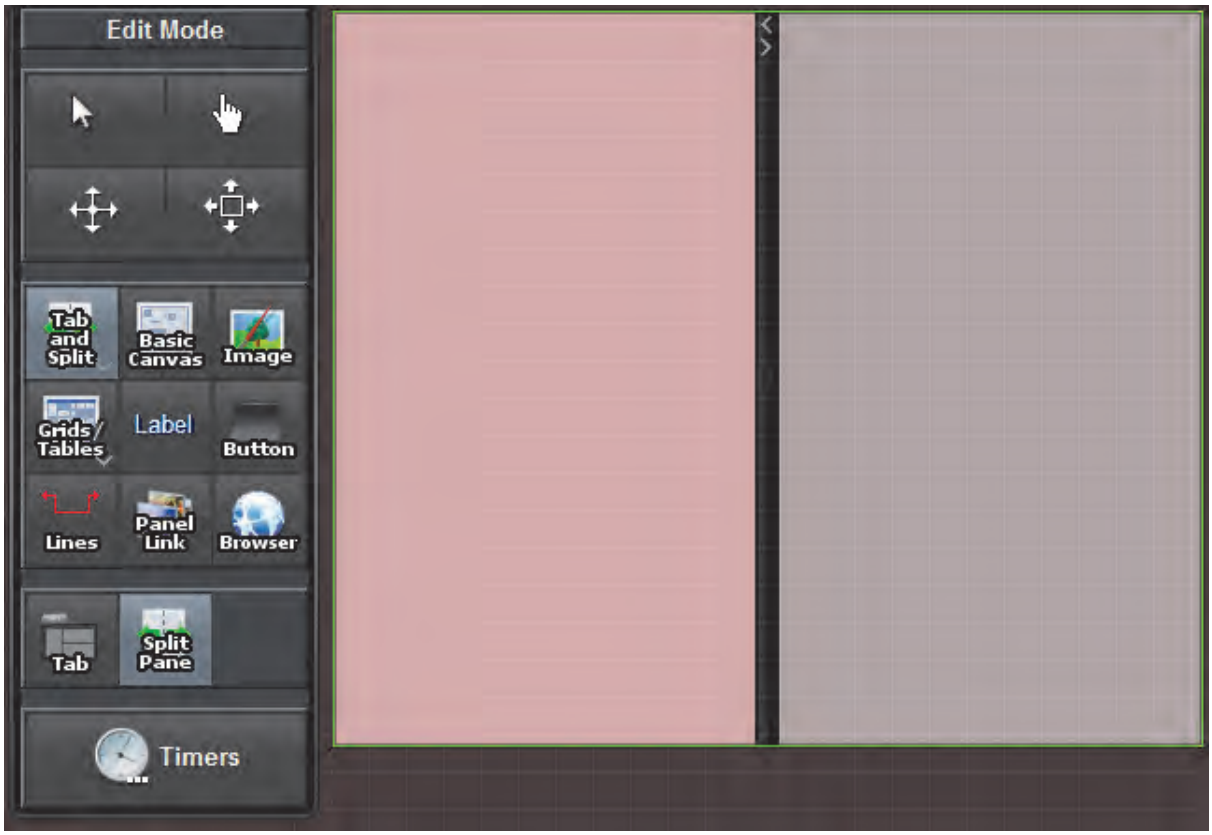



***Figure 5.17*** *- Adding a Split Panel*

**7.** If you want to split a panel, click it and then specify the orientation and position of the new split bar.

### Editing Split Panel Attributes

After you create a split panel, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For split panels, the Edit Component window may contain the following tabs:

- **Split Attributes Tab** — For more information, see "**Split Attributes Tab**" on page 5–73.
- **Dropspot Attributes Tab** — For more information, see "**Dropspot Attributes Tab**" on page 5–62.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Tables

A table is a grid of cells to which you can add other components. Each cell is a dropspot. Tables enable you to neatly arrange small components such as buttons and status indicators.

### Tips about Tables

- **Deleting table cells** — If you delete all the cells in a row or column, the remaining cells expand to fill the table area.
- **Table of Buttons** — When you create a table, you can specify that it be filled with buttons. This is useful for creating a control panel with perfectly identical buttons. After you create the table of buttons, you can edit them individually to define their names, button types, tasks, etc. For more information, see "**Editing Button Attributes:**" on page 5–32.
- **Selecting a table or table cell** — As you hover over a table, a border appears, indicating which component is selected. The component can be the entire table or an individual table cell.
- **Table formatting** — Right-click a table cell to access the following formatting options:
  - **Set all cell sizes to selection** — Makes all cells the same size. If you display parameters in table cells, the cells resize to fit the parameter text. This option makes them all the same size.
  - **Insets** — Provides options for adding padding to individual cells or all cells. You can also remove padding from individual cells or all cells.
  - **Add Row(s)** — Adds one or more empty rows to the bottom of the table. You can choose whether the table stays the same size (cells shrink), or grows to accommodate the new row(s).
  - **Duplicate Row** — Creates one or more duplicates of the current row, including all cell contents. You can choose whether the table stays the same size (cells shrink), or grows to accommodate the new row(s).
  - **Remove Row(s)** — Deletes the current row.
  - **Lock all proportions** — The table and its cells automatically scale as the DashBoard window is resized. This option is useful for accommodating different screen sizes and resolutions.
  - **Unlock all proportions** — The table and its cells maintain their current sizes and shapes. This option is useful for ensuring a consistent visual display.
  - **Delete** — Deletes the selected cell. Cells to the right of the deleted cell shift left.

  **Note:** There are many other formatting options available through the Edit Component window. Double-click a cell to edit its properties.

### To create a table:

1. On the **Edit Mode** toolbar, click the **Table** button.

   **Tip:** If the **Table** button is not visible, click the **Grids/Tables** button to reveal the **Table** button.

2. Drag a box on the CustomPanel to define the table area.

   The **Insert into Component** dialog appears.

3. Specify the number of rows and columns in the table.
4. If you want all the table cells to contain buttons, select the **Fill with buttons** box.
5. If the table is going to be populated by parameter values and you want to limit the number of columns in the table, set **Max Elements Per Row**.

   For example, this option is useful if you want to create a table of buttons, each of which includes a choice as defined in a parameter with nine values. Create a one-cell table and set **Max Elements Per Row** to **3**. Drag the parameter onto the table, setting it as a choice list, with the **Keep returned elements together** option unselected. The table will have three rows of three buttons, each of which contains one of the nine choices defined in the parameter.

6. Click **OK**.

   The table appears. **Figure 5.18** shows a table with three rows and three columns.

   Each cell contains an **X** to indicate that it is empty.

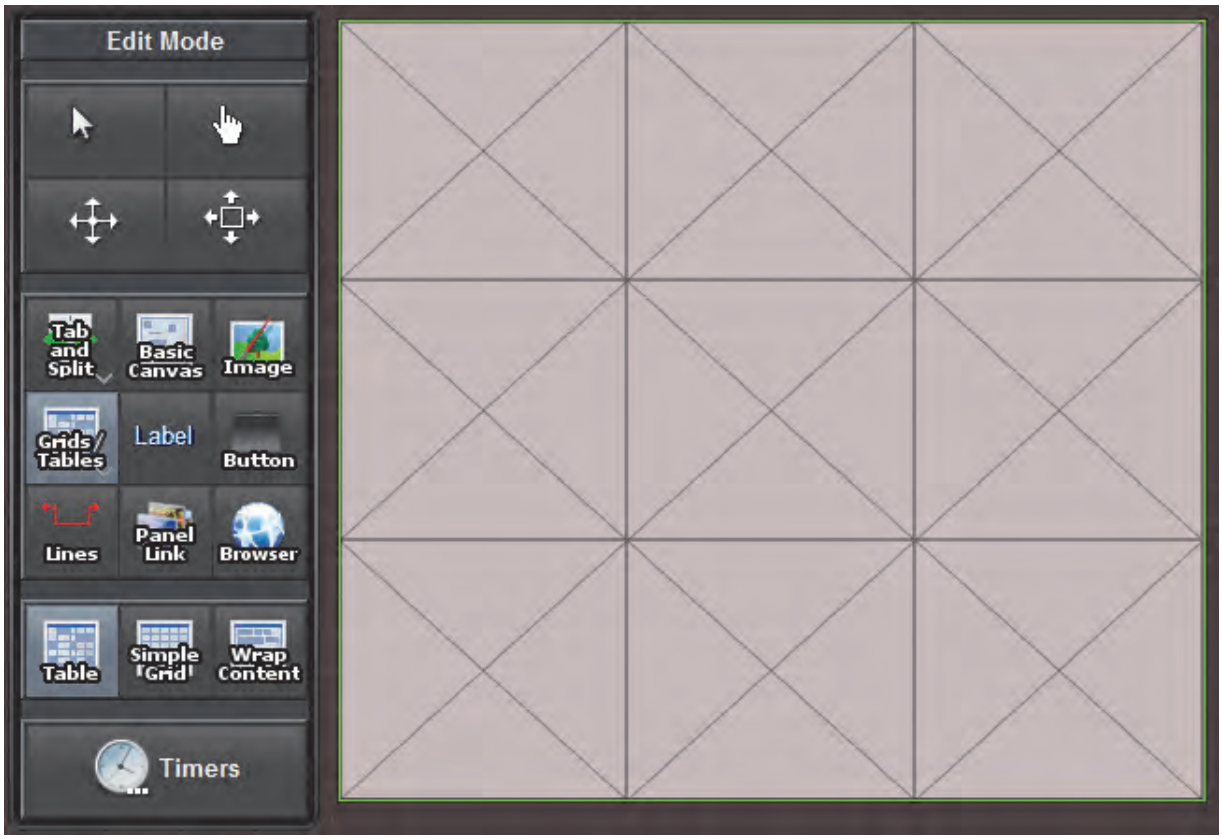



***Figure 5.18*** *- Adding a Table*

### Editing Table Attributes

After you create a table, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For tables and table cells, the Edit Component window may contain the following tabs:

- **Table Attributes Tab** — For more information, see "**Table Attributes Tab**" on page 5–78.
- **Container Attributes Tab** — For more information, see "**Container Attributes Tab**" on page 5–61.
- **Dropspot Attributes Tab** — For more information, see "**Dropspot Attributes Tab**" on page 5–62.
- **Tr Attributes Tab** — For more information, see "**Tr Attributes Tab**" on page 5–82.
- **Abs Attributes Tab** — For more information, see "**Abs Attributes Tab**" on page 5–58.
- **Table Cell Attributes Tab** — For more information, see "**Table Cell Attributes Tab**" on page 5–79.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Simple Grids

A simple grid is like a table, but with all cells the same size. Each cell is a dropspot into which you can insert other components.

When you create a simple grid, you specify the number of rows and/or columns. If you insert more components than there are cells in the grid, additional columns are created.

We recommend you specify the number of rows and columns when you create the simple grid. Otherwise, the simple grid has only one row and it divides into equal-width columns as you add components.

**To create a simple grid:**

1. On the **Edit Mode** toolbar, click the **Simple Grid** button.

   **Tip:** If the **Simple Grid** button is not visible, click the **Grids/Tables** button to reveal the **Simple Grid** button.

2. Drag a box on the CustomPanel to define the grid area.

   The **Insert into Component** dialog appears.

3. Beside **Rows** and/or **Columns**, select **Override Default**, and then specify the number of rows and/or columns you want in the simple grid.

4. Click **OK**.

   The simple grid appears. The rows and columns are not apparent until you insert components.

### Editing Simple Grid Attributes

After you create a simple grid, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For simple grids, the Edit Component window may contain the following tabs:

- **Simplegrid Attributes Tab** — For more information, see "**Simplegrid Attributes Tab**" on page 5–71.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Flow Containers (Wrap Content)

A flow container is like a table, but without a predefined number of rows and columns.

When you create a flow container, you can specify whether to keep all components widths and/or heights the same.

As you add components to a flow container, each is added to the right of the previous one. When a row is filled, additional components appear in the next row.

**To create a flow container (wrap content):**

1. On the **Edit Mode** toolbar, click the **Wrap Content** button.

   **Tip:** If the **Wrap Content** button is not visible, click the **Grids/Tables** button to reveal the **Wrap Content** button.

2. Drag a box on the CustomPanel to define the flow container area.

   The **Insert into Component** dialog appears.

3. If you want the components to be neatly aligned along the right or left edge of the container, or centered within the container, specify the **Horizontal Alignment** accordingly.

4. If you want the widths of all components in the container to be the same, select **Keep all widths the same**.

   All component widths will match the width of the widest one.

5. If you want the heights of all components in the container to be the same, select **Keep all heights the same.**

   All component heights will match the height of the tallest one.

6. If **Keep all widths the same** is selected, and you want all the components to fill a single row if possible, select **Fill single line if possible**.

   If the components widths are small enough that the components can all fit on one row with extra space, the widths are expanded to fill the row.

7. Click **OK**.

   The flow container appears. No rows or columns are apparent until you insert components.

### Editing Flow Container Attributes

After you create a flow container, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For flow containers, the Edit Component window may contain the following tabs:

- **Flow Attributes Tab** — For more information, see "**Flow Attributes Tab**" on page 5–63.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Labels

Labels are blocks of stand-alone text. Labels can be used as headers or banners anywhere on the CustomPanel, positioned beside components to provide descriptions of the component function, or added anywhere to provide additional information.

You can also assign tasks to a label, so that PanelBuilder performs the tasks when a user clicks the label. For more information, see "**Assigning Tasks to Buttons, Labels, and Timers**" on page 5–43.

**To create a label:**

1. On the **Edit Mode** toolbar, click the **Label** button.
2. Drag a box on the panel to define the label area.

   The **Insert into Component** dialog appears.
3. In the **Name** box, type the text you want to appear on the label.
4. In the **ID** box, specify an ID (optional).

   IDs are used in scripts to refer to objects. If no scripting is required, you do not need to specify an ID.
5. In the **Text Alignment** box, specify how to align the text within its box.
6. If you want to format the text as a banner, select the **Header** box.

   The background of the label area is automatically set to blue and the text is set to white.
7. Click **OK**.

   The label appears. **Figure 5.19** shows a label with a red background.

   **Tip:** By default labels have no borders. You can edit the border settings to create a border around the label.

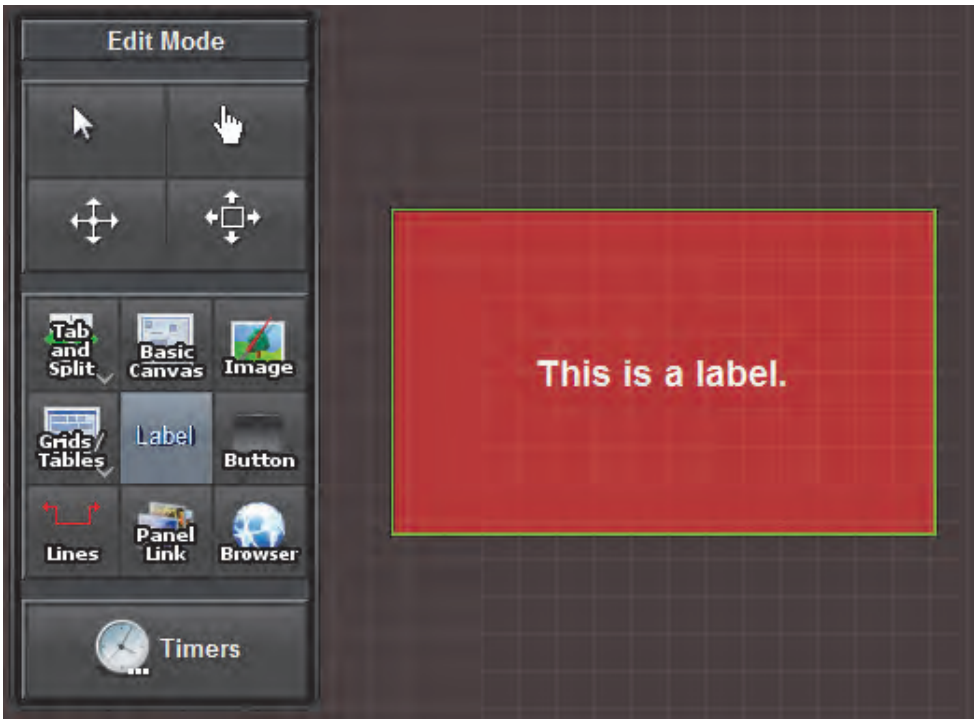



***Figure 5.19*** *- Adding a Label*

### Editing Label Attributes

After you create a label, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For labels, the Edit Component window contains the following tabs:

- **Label Attributes Tab** — For more information, see "**Label Attributes Tab**" on page 5–64.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Links to Device Editors or Other CustomPanels

You create a link that, when clicked by the user, automatically opens a device editor or another CustomPanel in the Device View.

**Tips About Links:**

- You can create link areas over an image canvas, to add links to a picture. When editing the link, on the **Tag Attributes** tab set **Button Style** to **Label**. On the **Style** tab, set **Background Fill** to **None**.

**To create a link:**

1. On the **Edit Mode** toolbar, click the **Panel Link** button.
2. Drag a box on the panel to define the link area.

   The **Insert into Component** dialog appears.
3. If you want to link to a device editor, in the **All Connections** list, double-click the device node to which you want to link.

   **Tip:** The **All Connections** list shows the contents of your DashBoard client's Tree View.

4. If you want to link to a CustomPanel, do one of the following:
   - › In the **File Navigator** list, double-click the CustomPanel (.grid file) to which you want to link.
   - › In the **Local File** area, browse to the CustomPanel (.grid file) to which you want to link.
5. Specify whether the link should appear as a button or as a label.
6. Click **OK**.

   The link button or label appears.

   **Tip:** By default there is no text on link buttons or labels, and link labels have no borders. This is useful for creating invisible link areas. If you want the link to be visible, you can edit the link name to add text, and edit the border settings to create a border.

### Editing Link Attributes

After you create a link, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For links, the Edit Component window contains the following tabs:

- **Tag Attributes Tab** — For more information, see "**Tag Attributes Tab**" on page 5–80.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Buttons

Buttons are controls you can add to CustomPanels to enable users to send commands or perform tasks.

This section describes how to create buttons on CustomPanels. For information about how to configure buttons to perform specific tasks, see "**Assigning Tasks to Buttons, Labels, and Timers**" on page 5–43.

You can create single buttons, or groups of buttons (table of buttons).

**To create a single button:**

1. On the **Edit Mode** toolbar, click the **Button** button.
2. Drag a box on the panel to define the button area.

   The **Insert into Component** dialog appears.
3. In the **Name** box, type the text you want to appear on or beside the button.
4. In the **ID** box, specify an ID (optional).

   IDs are used in scripts to refer to objects. If no scripting is required, you do not need to specify an ID.
5. In the **Type** list, click a button type:
   - **Push** — When clicked, PanelBuilder performs the assigned tasks.

     The visual appearance of the button changes momentarily while it is being clicked, and then reverts to its default appearance when it is released.
   - **Toggle** — When clicked, switches between two states (ON and OFF).

     When the button changes state, PanelBuilder performs the assigned tasks. Alternatively, you can create a script that detects the state change and performs tasks based on which state has been activated.

**Tip:** The visual appearance (style) of the button can be different for each of three states (Default, Toggle On, and Toggle Off). Toggle button styles are defined on subtabs of the Style tab, in the Edit Component window.

- **Checkbox** — When clicked, switches between two states (ON and OFF).

  When the button is ON, the check box contains a check mark. When the button is OFF, the check box is empty.

  When the button changes state, PanelBuilder performs the assigned tasks. Alternatively, you can create a script that detects the state change and performs tasks based on which state has been activated.

  **Tip:** The visual appearance (style) of the button can be different for each of three states (Default, Toggle On, and Toggle Off). Checkbox button styles are defined on subtabs of the Style tab, in the Edit Component window.

- **Radio** — When clicked, switches between two states (ON and OFF).

  When the button is ON, the round button is filled. When the button is OFF, the round button is empty. When the button changes state, PanelBuilder performs the assigned tasks. Alternatively, you can create a script that detects the state change and performs tasks based on which state has been activated.

6. If you do not want the button to have a three-dimensional visual effect, select the **Flat Look** box.
7. Use the options in the **Task** area to configure the button to perform one or more tasks when the user selects it. For more information, see "**Assigning Tasks to Buttons, Labels, and Timers**" on page 5–43.
8. Click **OK**.

The button appears. **Figure 5.20** shows the toolbar and various types of buttons.

***Figure 5.20*** *- Adding Buttons*

**To create a group of buttons:**

1. Create a table and select the **Fill with buttons** option.

   For more information, see "**Tables**" on page 5–25.

2. Edit the buttons individually to define their names, button types, tasks, etc.

### Editing Button Attributes:

After you create a button, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it.

For single buttons and grouped buttons, the Edit Component window may contain the following tabs:

- **Button Attributes Tab** — For more information, see "**Button Attributes Tab**" on page 5–60.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.
- **Table Cell Attributes Tab** — For more information, see "**Table Cell Attributes Tab**" on page 5–79.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Line Segments

You can create line segments on CustomPanels.

When you create a line segment, you define the area it occupies. By default, the line extends from the top left corner of the area to the bottom right corner, and has an arrow at the bottom right end. The line has three nodes: one at the start, one where it bends, and one at the end. You can move the nodes, and also insert additional nodes to extend the line path.

**To create a line segment:**

1. On the **Edit Mode** toolbar, click the **Insert line segments** button.
2. Drag a box on the panel to define the line segment area.

   A line appears, with three nodes, as shown in **Figure 5.21**.

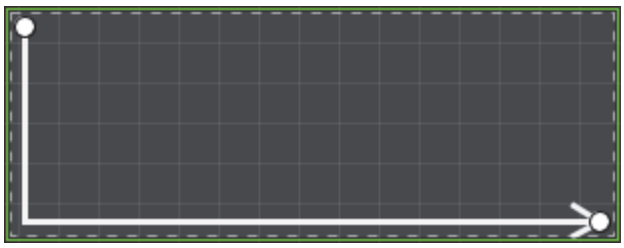


***Figure 5.21*** *- Creating a Line Segment*

3. Modify the line segment as desired, in any of the following ways:
   - Change the size and proportions of the line by resizing its container:

     On the **Edit Mode** toolbar, click the **Resize components** button, and then drag the sides and corners of the line segment box.
   - Select a node:

     Point at a node and then press **Ctrl+click**.
   - Reposition a node:

     Point at the node, press and hold **Ctrl+click**, and drag the node.

- Add a node at the end of the line:

  Point to where you want the new node, and the press **Ctrl+double-click**. The line extends to the new node.

- Add a node along the line:

  Point to a node adjacent to where you want the new node, and then press **Ctrl+double-click**. The new node is created in the same position. Reposition the new node.

- Delete a node:

  Point at the node, press **Ctrl+click** to select the node, and then press **Delete**.

- Delete the entire line segment:

  On the **Edit Mode** toolbar, click the **Select/Drag Components** button, click the line segment, and then press **Delete**.

**To further modify a line segment using the Edit Component dialog:**

1. In the **Edit Mode** toolbar, click the **Select/Drag components** button, double-click the line segment to open the **Edit Component** dialog, and then click the **Source** tab.

   The source code that defines the line segment appears. A line segment consists of a lines element which contains two or more point elements. The line extends from the first point through each of the subsequent points.

2. Edit general line characteristics as follows:

   - To allow the line to travel diagonally from point to point, change the **`diagonals`** attribute to true.
   - To show arrows at the start and/or end of the line, set the **`startarrow`** and/or **`endarrow`** attributes to **`true`**, respectively.
   - Specify whether the horizontal (**`x`**) and vertical (**`y`**) position of points is defined in pixels or as a percentage of the line segment area:
     - For pixels, set the **`percent`** attribute to **`true`**.
     - For percentage, set the **`percent`** attribute to **`false`**.

3. Edit point characteristics as follows:

   - To move a point, edit the **`point`** element's horizontal (**`x`**) and vertical (**`y`**) position values.
   - To add a point, add a **`point`** element and define its horizontal (**`x`**) and vertical (**`y`**) position values.
   - To create a closed shape, make the first and last **`point`** elements identical.

4. To change the style of the line segment, click the **Style** tab and then change settings as desired.

5. To change the position or anchoring of the line segment area (container), click the **Position/Stretch attribute**s tab, and then change settings as desired.

6. When you are finished editing the line segment, click **Apply and Close**.

## Web Browser Instances

You can add active web browser instances to your CustomPanel to display fully-interactive web pages. Web browser instances are like canvases, which you can overlay with other components.

**Tips about Web Browser Instances:**

- **Web pages are not displayed in Edit Mode** — After you add a web browser instance, you must exit Edit Mode to see the actual web page.
- **Not a complete browser** — Web pages are fully functional, but do not include typical browser features such as an address bar or forward and back buttons.

**To add a web browser instance:**

1. On the **Edit Mode** toolbar, click the **Insert web browser** button.
2. Drag a box on the panel to define the web browser area.

   The **Insert into Component** dialog appears.
3. In the **URL** box, type the address of the web page you want to display in the browser.

   For example, to display the Ross Video website, type `http://www.rossvideo.com`.
4. Click **OK**.

   The browser instance appears.

### Editing Web Browser Attributes

After you add a web browser instance, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For web browser canvases, the Edit Component window contains the following tabs:

- **Browser Attributes Tab** — For more information, see "**Browser Attributes Tab**" on page 5–59.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

# Adding Data-Backed Components

If the CustomPanel is associated with an XML data source file, the Edit Mode toolbar includes additional buttons for creating objects that display and/or manipulate data parameters.

This section includes generic steps for creating data-backed objects such as static or editable displays of parameter data, static labels, editable text areas, option choice controls, numeric choice controls, or toggle choice controls.

**For More Information on...**

- For detailed information about each type of data-backed object, see the corresponding section:
  - "**Parameter Displays**" on page 5–35
  - "**Data-Backed Labels**" on page 5–36
  - "**Editable Text Areas**" on page 5–36
  - "**Option Choice Controls**" on page 5–37
  - "**Numeric Choice Controls**" on page 5–37
  - "**Toggle Choice Controls**" on page 5–38
- For information about customizing components by editing their attributes, see "**Editing Components**" on page 5–53.

## Parameter Displays

You can display a parameter value, which users can change.

You can also associate one or more tasks with the displayed parameter, so that when the parameter value changes, the tasks are performed.

**To display a parameter value:**

1. On the **Edit Mode** toolbar, click the **Display or edit a parameter backed by your data source** button.

   **Tip:** If the button is not visible, click the **Param** button to reveal the buttons used for adding data-backed components.

2. Drag a box on the panel to define the area.

   The **Insert into Component** dialog box appears.

3. In the **Select Parameter** area, select the parameter you want to display.

4. Specify whether you want to include the parameter name.

   When selected, the name of the parameter, as defined by the data source, is also displayed.

5. If the parameter may return multiple items, specify whether you want them kept together.

   - When selected, returned elements can only be modified as a group, and are displayed together neatly. For example, if placed on an absolute position canvas, they do not overlap.
   - When not selected, returned elements can be individually modified. For example, you can apply different style options to each element, or position them in separate table cells.

6. Click **OK**.

   The parameter appears on the CustomPanel.

### Editing Displayed Parameters

After you display a parameter, you can customize the display using the Edit Component window.

To access the Edit Component window, select the component and double-click it. For displayed parameters, the Edit Component window contains the following tabs:

- **Param Attributes Tab** — For more information, see "**Param Attributes Tab**" on page 5–68.

  **Tip:** In the **Tasks** area of the **Param Attributes** tab, you can associate one or more tasks with the displayed parameter, so that when the parameter value changes, the tasks are performed.

- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Data-Backed Labels

You can insert a text label showing data from a parameter. You can select an existing parameter, or create a new one. Data-backed labels are read-only to the user.

**To create a data-backed label:**

1. On the **Edit Mode** toolbar, click the **Label** button that is partially yellow.

   **Tip:** If the button is not visible, click the **Param** button to reveal the buttons used for adding data-backed components.

2. Drag a box on the panel to define the label area.

   The **Insert into Component** dialog box appears.

3. In the **Insert Label** area, select an existing parameter, or create a new one.
4. In the **Display Type** area, select a display option.
5. Click **OK**.

   The label appears on the CustomPanel.

### Editing Data-Backed Labels

After you add a data-backed label, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For data-backed labels, the Edit Component window contains the following tabs:

- **Param Attributes Tab** — For more information, see "**Param Attributes Tab**" on page 5–68.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Editable Text Areas

You can add a text area backed by a data source. Depending on how you choose to display the text area, it can be editable or not, can be shown as a selectable list, or appear as a status dot.

You can format the text area using the options provided by the data source library. The entered data is stored in the associated parameter.

**To create an editable text area:**

1. On the **Edit Mode** toolbar, click the **Insert an editable text area backed by your data source** button.

   **Tip:** If the button is not visible, click the **Param** button to reveal the buttons used for adding data-backed components.

2. Drag a box on the panel to define the text area.

   The **Insert into Component** dialog box appears.

3. In the **Insert String** area, select an existing parameter, or create a new one.
4. In the **Display Type** area, select a display option.
5. Click **OK**.

   The text area appears on the CustomPanel.

### Editing Text Area Attributes

After you add a text area, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For editable text areas, the Edit Component window contains the following tabs:

- **Param Attributes Tab** — For more information, see "**Param Attributes Tab**" on page 5–68.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Option Choice Controls

You can add controls that enable users to select from a pre-determined list of options. Choices can be shown in a text list, drop-down menu, list of rectangular buttons, or as a list of radio buttons.

**To create an option choice control:**

1. On the **Edit Mode** toolbar, click the **Insert a choice (list, toggle buttons, or, radio buttons) backed by your data source** button.

   **Tip:** If the button is not visible, click the **Param** button to reveal the buttons used for adding data-backed components.

2. Drag a box on the panel to define the area for the control.

   The **Insert into Component** dialog box appears.

3. In the **Insert Choice** area, select an existing parameter, or create a new one.

4. In the **Display Type** area, select a display option.

5. Click **OK**.

   The option choice control appears on the CustomPanel.

### Editing Option Choice Controls

After you add an option choice control, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For option choice controls, the Edit Component window contains the following tabs:

- **Param Attributes Tab** — For more information, see "**Param Attributes Tab**" on page 5–68.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Numeric Choice Controls

You can add a control that enables users to specify numeric values for a parameter.

**To create a numeric choice control:**

1. On the **Edit Mode** toolbar, click the **Insert a number (slider, counter, etc) backed by your data source** button.

   **Tip:** If the button is not visible, click the **Param** button to reveal the buttons used for adding data-backed components.

2. Drag a box on the panel to define the area for the control.

   The **Insert into Component** dialog box appears.

3. In the **Insert Number** area, select an existing parameter, or create a new one.

   If you create a new parameter, specify the range constraints (minimum value, maximum value, and step value).

4. In the **Display Type** area, select a display option.

5. Click **OK**.

   The numeric choice control appears on the CustomPanel.

### Editing Numeric Choice Controls

After you add a numeric choice control, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For numeric choice controls, the Edit Component window contains the following tabs:

- **Param Attributes Tab** — For more information, see "**Param Attributes Tab**" on page 5–68.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Toggle Choice Controls

Toggle choice controls enable the user to make a choice between two states. Choose between check boxes and toggle switches to customize how the user selects a state. The selected state data is stored in the associated parameter.

**To create a toggle choice control:**

1. On the **Edit Mode** toolbar, click the **Insert a Toggle Choice (checkbox or single toggle button) backed by your data source** button.

   **Tip:** If the button is not visible, click the **Param** button to reveal the buttons used for adding data-backed components.

2. Drag a box on the panel to define the area for the control.

   The **Insert into Component** dialog box appears.

3. In the **Insert Toggle** area, select an existing parameter, or create a new one.

   If you create a new parameter, you can change the values and names for the true and false values. The name can be displayed on the button or beside the check box.

4. In the **Display Type** area, select a display option.

5. Click **OK**.

   The toggle choice control appears on the CustomPanel.

### Editing Toggle Choice Controls

After you add a toggle choice control, you can customize it using the Edit Component window. To access the Edit Component window, select the component and double-click it. For toggle choice controls, the Edit Component window contains the following tabs:

- **Param Attributes Tab** — For more information, see "**Param Attributes Tab**" on page 5–68.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.

For more information about using the Edit Component window, see "**Editing Components**" on page 5–53.

## Creating a Row, Column, or Grid of Data-Backed Buttons

You can create a row, column, or grid of buttons based on a parameter. Users can click the buttons to change the parameter value.

**Figure 5.22** shows a row of buttons (top), a column of buttons (left), and a grid of buttons (right). In this example, the **three** button is selected. The **Toggle On** style for the parameter has been changed to make the button background red.

***Figure 5.22*** *- Tables of buttons, including a row (top), a column (left), and a grid (right)*

**To create a row, column, or grid of buttons based on a parameter:**

1. Ensure the parameter you want to use has the following characteristics:
   - Parameter **Type** is set to **Integer**
   - **Constraint** is set to **Choice Constraint**
   - **Constraint Value** box lists the text strings you want to show on the buttons
   - **Widget Hint** is set to **Toggle Buttons**
   - The **Initial Value** is set
2. Create a table with one row and one column.

For more information, see "**Tables**" on page 5–25.

3. In the **Edit Mode** toolbar, click the **Display or edit a parameter backed by your data source icon**, and then click the table cell.

   **Tip:** If the button is not visible, click the **Param** button to reveal the buttons used for adding data-backed components.

   The **Insert into Component** dialog appears.

4. Select the parameter, clear the check boxes at the bottom of the dialog, and then click **Ok**.
5. If you want to change the table to be a single column or a grid of buttons, edit the table, and on the **Container Attributes** tab, set the **Max Elements Per Row** to the number of columns you want.
6. If you want to set different styles for different button states, select the parameter in the **Component List**, and then edit the style settings on the **Toggle On** and **Toggle Off** tabs.
7. Resize and reposition the table as required.

## Timers

Timers control and display time information. They can be displayed on a CustomPanel, or can operate without being displayed. Timers can include scheduled tasks which run when the timer reaches specified values.

You can create the following types of timers:

- **Self** — starts and stops manually, and is independent of any other timer.
- **Simple clock** — matches the time clock of the local computer.
- **Time Until** — counts down the amount of time remaining until a future date and time.
- **Other Timer** — links this timer to another timer. Starting or stopping the linked timer also starts or stops this timer.

**Tips about Timers:**

- **Timers are not panel components** — To view or manipulate a timer in a CustomPanel, you must create the timer, and then create a panel component such as a data-backed label or button, associated with the timer.
- **Timers run tasks** — Each timer includes a repeat rate, which defines how often DashBoard performs the list of tasks associated with the timer. For example, if the repeat rate is 500 milliseconds, the tasks are performed twice per second.
- **Avoid frequent repeat rates** — Each time the timer repeats, it performs all the tasks on the task list. If you set the timer to repeat too frequently, such as once per video field or video frame, DashBoard may not be able to process the task requests quickly enough, and performance may lag or unpredictable results may occur.

  The repeat rate should generally be twice as frequent as the minimum display unit. For example, if the display is accurate to one second, the repeat rate should be 500 milliseconds.
- **Starting a timer** — You will need to create a button to start a timer if it is not using the computer clock. A self timer does not display a time until the timer is reset or started. It appears as a blank item.
- **Child timers** — The repeat rate for a child timer is controlled by its parent.

## Creating a Timer

When you create a timer, you specify how time values are reported, the timer type, start and stop times, and whether you want tasks to be performed at certain intervals during timer operation. Creating a timer does not automatically add it to your CustomPanel.

**To create a timer:**

1. On the **Edit Mode** toolbar, click the **Timers** button.

   The **Add/Edit Timers** dialog appears.

   **Tip:** The **Add/Edit Timers** dialog box lists all the timers associated with the current CustomPanel.

2. Click **Add New**.
3. In the **Timer ID** box, type a name for the timer.
4. In the **Display** box, do one of the following to specify the time format you want the timer to use:
   - Expand the **Display** list and double-click the format.
   - Type the format and then press **Enter**.

   **Tip:** To view descriptions of the time formatting symbols, click beside the **Display** list.
5. Click one of the following buttons to specify the type of timer you want:
   - **Manual** — Creates a timer with start and stop times, for counting up or down. You can specify positive or negative values.

     **Tip:** To create a countdown timer, set the start time later than to the stop time.
   - **Simple Clock** — Creates a timer that counts forward, matching the DashBoard computer's clock time.
   - **Count Time Until** — Allows you to specify a future date and time as a reference. The timer value is the amount of time before the reference time arrives.
   - **Other Timer** — Enables you to copy the properties of an existing timer.
6. If you want the timer to start one or more tasks, in the **Tasks** area, specify the tasks to perform.

   For more information about specifying tasks, see "**Assigning Tasks to Buttons, Labels, and Timers**" on page 5–43.
7. Click **Commit Changes** to create the timer and to add it to list of timers.
8. Click **Done**.

## Adding Timer Labels and Timer Control Buttons to a CustomPanel

After you create a timer, you can add a label to display the timer data. You can also add buttons to control the timer.

**To add a label that shows timer data:**

1. On the **Edit Mode** toolbar, click the **Insert a data-backed label** button.

   **Tip:** If the button is not visible, click the **Param** button to reveal the buttons used for adding data-backed components.
2. Drag a box on the panel to define the label area.

   The **Insert into Component** dialog appears.
3. Select **Create New Parameter**.
4. In the **Name** box, type a name for the timer label.
5. Select **Timer Value**.
6. In the **Timer Value** list, double-click the timer you want to add.
7. In the **Display Type** area, select a style for the timer label.

8. Click **OK**.

**To add start and stop buttons for your timer:**

1. Create a table containing two buttons.

   For detailed instructions, see "**To create a group of buttons:**" on page 5–32.

   **Tip:** The buttons don't have to be in a table. The table aligns them neatly.

2. Configure the first button as a **Start** button:

   a. Select from the **Edit Mode** toolbar.

   b. Double-click the first button.

      The **Edit Component: <button>** dialog appears.

   c. On the **Button Attributes** tab, type `Start` in the **Name** box.

   d. In the **Tasks** area, click **Add**.

   e. In the **Task Type** area, click **Timer Control**.

      The **Timer Control Editor** appears.

   f. In the **Timer** list, double-click the timer you want to control.

   g. In the **Action** list, click **Start Timer**.

   h. Click **OK**.

   i. Click **Apply and Close**.

3. Configure the second button as a **Stop** button by repeating **Steps 2a to 2i** for the second button, but with the following changes:

   - In **Step 2c**, type `Stop` in the **Name** field.
   - In **Step 2g**, click **Stop Timer** in the **Action** drop-down menu.
   - If you want the timer to reset when the **Stop** button is clicked, select the **Reset Timer** check box.

## Timer Control Functions

There are several timer control functions:

- **Start Timer** — starts the associated timer.

  If you also select the **Reset Timer** check box, the timer resets before starting.
- **Stop Timer** — stops the associated timer.

  If you also select the **Reset Timer** check box, the timer resets before stopping.
- **Reset Timer** — resets the associated timer.
- **Set Time** — sets the timer to a specific time.
- **Add/Remove Time** — adds or removes the specified amount of time to a timer. Positive values add time, negative values remove time.
- **Set Start Time** — sets the start time. The timer begins counting from this time.
- **Set Stop Time** — sets the stop time. The timer stops counting at this time.
- **Set Pattern** — sets the display pattern of the timer. You can specify a custom pattern or choose a predefined time format.

For information about how to link timers to buttons on a CustomPanel, see "**To add start and stop buttons for your timer:**" on page 5–42.

## Assigning Tasks to Buttons, Labels, and Timers

You can assign a list of one or more tasks to each button, label, or timer. When the user clicks a label or button, the associated tasks are performed. When a timer reaches its threshold value (repeat rate), its associated tasks are performed.

**Note:** For labels or buttons, you can also trigger task lists externally, through keyboard shortcuts or RossTalk messages. For more information, see "**Triggering Tasks Externally**" on page 5–50.

The types of tasks you can assign include the following:

- **ogScript** — runs an ogScript. You can also use the provided ogScript Editor to parse XML data.

  For more information about ogScript, see the ***ogScript Reference Guide (8351DR-006)***.
- **RossTalk Script** — sends a RossTalk command to a specific device to perform a task. Devices include Vision and Carbonite production switchers, and XPression graphics systems.

  For more information about the RossTalk protocol, see the document, ***RossTalk-Commands (4802DR-403)***.
- **CamBot** — sends a command to a specific Ross Robotics CamBot™ device to perform a task.
- **VDCP Command** — sends a VDCP command to a Ross BlackStorm Video Server, to perform a task.
- **PBus Command** — sends a PBus command to XPression, or to another device that uses the PBus protocol.
- **Timer Control** — performs a timer function, such as count up/down and stop.
- **Data Modification** — modifies parameter values as specified.

This section describes how to assign tasks, and includes the following topics:


- "**Assigning ogScript Tasks**" on page 5–43
- "**Assigning Pauses**" on page 5–44
- "**Assigning RossTalk Commands**" on page 5–44
- "**Assigning CamBot Commands**" on page 5–45
- "**Assigning VDCP Commands**" on page 5–46
- "**Assigning PBUS Commands**" on page 5–47
- "**Using the Global List**" on page 5–48
- "**Assigning Data Modification Tasks**" on page 5–48
- "**Assigning Timer Control Tasks**" on page 5–49
- "**Editing a Task**" on page 5–50


### Assigning ogScript Tasks

Ross ogScript is a programming language developed by Ross Video to interact with DashBoard-enabled devices. It is a subset of JavaScript, with additional PanelBuilder-specific functions added.

You can add advanced functionality and logic to CustomPanels by creating tasks that execute ogScript.

You can associate ogScript tasks with buttons, labels, parameters, timers, or listeners.

**To assign a ogScript task:**

1. Create the item with which you want to associate the task.
2. In edit mode, double-click the panel, to open the **Edit Component** window.
3. In the component tree, navigate to the item.
4. In the attribute editor area, open the **Attributes** tab for the item.

   **Tip:** The **Attributes** tab will be one of the following: **Button Attributes**, **Label Attributes**, **Param Attributes**, **Timer Attributes**, or **Listener Attributes**.

5. In the **Tasks** area, click **Add**.

   The **Add Task** dialog appears.

6. In the **Task Type** area, click the **ogScript** button.

   The **ogScript Editor** dialog appears.

7. Type or paste the ogScript code into the ogScript Editor.

8. Click **OK**.

   The ogScript task is added to the Tasks list.

## Assigning Pauses

You can add timed pauses between tasks in a task list.

**To add a pause to a task list:**

1. Open a timer or control that includes a task list.

2. In the **Tasks** area, click the task that you want the pause to follow.

3. Click **Add**.

   The **Add Task** dialog appears.

4. In the **Task Type** area, click the **Pause** button.

   The **Pause Editor** dialog appears.

5. Specify the duration of the pause.

   **Tip:** You can specify the unit of measure in time units (milliseconds, seconds, minutes), or frames (at 24 fps, 25 fps, 30 fps, 50 fps, or 60 fps).

   **Important Note:** Pauses are run on the DashBoard computer and will not be frame accurate. They will **roughly** be of the duration specified.

6. Click **OK**.

   The pause is added to the Tasks list.

## Assigning RossTalk Commands

RossTalk is an Ethernet-based protocol that allows the control of Ross Video devices using simple commands. Refer to the documentation that came with your device to verify which commands it supports.

You can associate commands with buttons, labels, parameters, timers, or listeners.

**To assign a RossTalk command:**

1. Create the item with which you want to associate the command.

2. In edit mode, double-click the panel, to open the **Edit Component** window.

3. In the component tree, navigate to the item.

4. In the attribute editor area, open the **Attributes** tab for the item.

   **Tip:** The **Attributes** tab will be one of the following: **Button Attributes**, **Label Attributes**, **Param Attributes**, **Timer Attributes**, or **Listener Attributes**.

5. In the **Tasks** area, click **Add**.

   The **Add Task** dialog appears.

6. In the **Task Type** area, click the **RossTalk** button.

   The **RossTalk Editor** dialog appears.

7. Configure the connection to the device. This is the device to which the command will be sent when the task runs. Do one of the following:

   - Specify the **Host** IP address and **Port** number of the device.

     The default port for RossTalk commands is port 7788.

   - Select a connection from the **Connection** list. For information on adding connections to the Connection list, refer to the section “Using the Global List” on page 5-48.

8. In the **Command** list, double-click the command you want to send.

   **Tip:** The **Command** list shows all commands. Some may not be applicable to the device you are controlling.

   **Tip:** The **Command** box is yellow until you provide the IP address of the device (see step **7** on page 5-45).

9. If additional command parameter boxes appear below the **Command** box, specify parameter values as required.

10. In the **Callback Function** area, type any commands to be executed after the RossTalk command is completed.

    **Note:** Commands that accept a callback do so because they run asynchronously. Callback commands are not guaranteed to execute before the next task. They are cued and run as soon as possible.

11. Click **OK** to apply your changes and to close the **Add Task** dialog.

12. Click **Apply Changes** or **Apply and Close**.

## Assigning CamBot Commands

This section describes how to assign CamBot commands. You can associate commands with buttons, labels, parameters, timers, or listeners.

**To assign a CamBot command:**

1. Create the item with which you want to associate the command.

2. In edit mode, double-click the panel, to open the **Edit Component** window.

3. In the component tree, navigate to the item.

4. In the attribute editor area, open the **Attributes** tab for the item.

   **Tip:** The **Attributes** tab will be one of the following: **Button Attributes**, **Label Attributes**, **Param Attributes**, **Timer Attributes**, or **Listener Attributes**.

5. In the **Tasks** area, click **Add**.

   The **Add Task** dialog appears.

6. In the **Task Type** area, click the **CamBot** button.

   The **CamBot Editor** dialog appears.

7. Configure the connection to the CamBot control computer. This is the CamBot control computer to which the command will be sent when the task runs. Do one of the following:

   - Specify the **Host IP** address and **Port** number of the CamBot control computer.

     The default port number is 2050.

   - Select a connection from the **Connection** list. For information on adding connections to the Connection list, refer to the section “Using the Global List” on page 5-48.

8. In the **Command** list, double-click the command you want to send.

   **Tip:** The **Command** box is yellow until you provide the IP address of the device (see step **7** on page 5-45).

9. If additional command parameter boxes appear below the **Command** box, specify parameter values as required.

10. In the **Callback Function** area, type any commands to be executed after the CamBot command is completed.

    **Note:** Commands that accept a callback do so because they run asynchronously. Callback commands are not guaranteed to execute before the next task. They are cued and run as soon as possible.

11. Click **OK** to apply your changes and to close the **Add Task** dialog.

12. Click **Apply Changes** or **Apply and Close**.

## Assigning VDCP Commands

This section describes how to send Video Disk Control Protocol (VDCP) commands from DashBoard to your Ross BlackStorm Video Server.

You can associate commands with buttons, labels, parameters, timers, or listeners.

**To assign a VDCP command:**

1. Create the item with which you want to associate the command.

2. In edit mode, double-click the panel, to open the **Edit Component** window.

3. In the component tree, navigate to the item.

4. In the attribute editor area, open the **Attributes** tab for the item.

    **Tip:** The **Attributes** tab will be one of the following: **Button Attributes**, **Label Attributes**, **Param Attributes**, **Timer Attributes**, or **Listener Attributes**.

5. In the **Tasks** area, click **Add**.

    The **Add Task** dialog appears.

6. In the **Task Type** area, click the **VDCP** button.

    The **VDCP Editor** dialog appears.

7. Configure the connection to the device. This is the device to which the command will be sent when the task runs. Do one of the following:

    - Specify the **Host** IP address and **Port** number of the device.
    - Select a connection from the **Connection** list. For information on adding connections to the Connection list, refer to the section “Using the Global List” on page 5-48.

8. In the **Command** list, double-click the command you want to send.

    **Tip:** The **Command** list shows all commands. Some may not be applicable to the device you are controlling.

9. **Tip:** The **Command** box is yellow until you provide the IP address of the device (see step **7** on page 5-46).

10. If additional command parameter boxes appear below the **Command** box, specify parameter values as required.

11. In the **Callback Function** area, type any commands to be executed after the VDCP command is completed.

    **Note:** Commands that accept a callback do so because they run asynchronously. Callback commands are not guaranteed to execute before the next task. They are cued and run as soon as possible.

12. Click **OK** to apply your changes and to close the **Add Task** dialog.

13. Click **Apply Changes** or **Apply and Close**.

## Assigning PBUS Commands

This section describes how to assign Peripheral BUS (PBUS) commands. You can send PBUS commands to control Ross Video XPression and other PBUS-enabled devices.

For detailed information about using PBUS with XPression, see the document, ***PBus on XPression.pdf***, which is available on your XPression system in the **C:\archives** directory.

You can associate commands with buttons, labels, parameters, timers, or listeners.

**To assign a PBUS command:**

1. Create the item with which you want to associate the command.
2. In edit mode, double-click the panel, to open the **Edit Component** window.
3. In the component tree, navigate to the item.
4. In the attribute editor area, open the **Attributes** tab for the item.

   **Tip:** The **Attributes** tab will be one of the following: **Button Attributes**, **Label Attributes**, **Param Attributes**, **Timer Attributes**, or **Listener Attributes**.
5. In the **Tasks** area, click **Add**.

   The **Add Task** dialog appears.
6. In the **Task Type** area, click the **PBus** button.

   The **PBus Editor** dialog appears.
7. Configure the connection to the device. This is the device to which the command will be sent when the task runs. Do one of the following:
   - Specify the **Host** IP address and **Port** number of the device.
   - Select a connection from the **Connection** list. For information on adding connections to the Connection list, refer to the section “Using the Global List” on page 5-48.
8. In the **Device** list, select the device to which you want to send the command.
9. In the **Command** list, double-click the command you want to send:
   - **Learn** — Commands the device to capture and store its current status. The device normally stores whatever data it requires to return to its current state.
   - **Recall** — Commands the device to return to a previously-stored state.
   - **Trigger** — Commands the device to execute an event which has been previously prepared. For example, data for a credit crawl can be read in advance, and then the trigger command executes the credit crawl.

   **Tip:** The **Command** box is yellow until you provide the IP address of the device.
10. In the **Register** list, type the register number, which is used as an identifier for the learned status. The register is specified when learning the status and for recalling the status.
11. In the **Callback Function** area, type any commands to be executed after the PBUS command is completed.

    **Note:** Commands that accept a callback do so because they run asynchronously. Callback commands are not guaranteed to execute before the next task. They are cued and run as soon as possible.
12. Click **OK** to apply your changes and to close the **Add Task** dialog.
13. Click **Apply Changes** or **Apply and Close**.

## Using the Global List

The Global List allows you to add a device connection to a list that enables easy association of commands with a similar device that uses a different IP address. This is helpful if you have multiple networks of devices or if you are moving around from different locations with a custom panel.

**To add a device connection to the Global List:**

1. Open the **Edit Component** dialog for a component (button, label, parameter, or listener).
2. Select the associated **Attributes** tab.
3. Click **Add**.

   The **Add Task** dialog opens.
4. Specify the **Host** IP address and **Port** number of the device.
5. Click the **Add Host/Port To Global List** button.

   The **Input** dialog opens.
6. Type a name for the host/port and click **OK**.

   The host/port appears in the **Connection** list.

**To assign a device connection using the global list:**

1. Open the **Edit Component** dialog for a a component (button, label, parameter, or listener).
1. Select the associated **Attributes** tab.
2. Click **Edit**.

   The **Edit Task** dialog opens.
3. Select a host/name from the **Connection** list.
4. Click **OK**.

## Assigning Data Modification Tasks

You can create a task that modifies parameter values.

**For More Information on...**

- parameters, refer to the section, "**Parameters and Data Sources**" on page 5–87.

You can associate tasks with buttons, labels, parameters, timers, or listeners.

**To assign a data modification task:**

1. Create the item with which you want to associate the task.
2. In edit mode, double-click the panel, to open the **Edit Component** window.
3. In the component tree, navigate to the item.
4. In the attribute editor area, open the **Attributes** tab for the item.

   **Tip:** The **Attributes** tab will be one of the following: **Button Attributes**, **Label Attributes**, **Param Attributes**, **Timer Attributes**, or **Listener Attributes**.
5. In the **Tasks** area, click **Add**.

   The **Add Task** dialog appears.
6. In the **Task Type** area, click the **Data Modification** button.

   The **Data Modification Editor** dialog appears.
7. In the list of parameters, select the parameter that the task will modify.

8. Choose to either set the parameter to a specific value or to increment / decrement it by a specific amount whenever the user selects it on the CustomPanel. Choose from the following:

   - **Set to value** — Specifies the value of the parameter value.
   - **Increment/Decrement Value** — Increases the parameter value by the specified value. To decrement the parameter value, specify a negative value.
   - **Set to event property** — Changes the parameter to the specified state.
   - **Set to script** — Allows you to modify the parameter value through a small piece of ogScript.

     For example, to set the parameter's value to the value of parameter 0x02 subtracted from parameter 0x01, you would type `params.getValue(0x01, 0) - params.getValue(0x02)`.

9. Click **OK** to apply your changes and close the **Add Task** dialog.
10. Click **Apply Changes** or **Apply and Close**.

## Assigning Timer Control Tasks

This section describes how to assign tasks to a timer.

**To assign a task to a timer:**

1. Create the timer, as described in the section, "**To create a timer:**" on page 5–41.
2. On the **Edit Mode** toolbar, click the **Timers** button.

   The **Add/Edit Timers** dialog appears.

   **Tip:** The Add/Edit Timers dialog lists all the timers associated with the current CustomPanel.
3. In the list of timers, click the timer to which you want to assign a task.

   The Timer Properties area shows settings for the timer. The Tasks area shows all task currently assigned to the timer. Some tasks are automatically created
4. In the **Repeat Rate** boxes, specify how often the list of tasks is to be performed.
5. In the **Tasks** area, click the **Add** button.

   The Add Task dialog appears.
6. In the **Task Type** area, click the type of task you want to associate with the timer.

   The Editor area updates according to the task type you selected.
7. Define the task using the **Editor** area of the **Add Task** dialog.

   For more information about defining the task, see one of the following:

   › For ogScript tasks, see Steps 5 to 8 in "**Assigning ogScript Tasks**" on page 5–43
   › For RossTalk tasks, see Steps 5 to 12 in "**Assigning RossTalk Commands**" on page 5–44
   › For CamBot tasks, see Steps 5 to 12 in "**Assigning CamBot Commands**" on page 5–45
   › For VDCP tasks, see Steps 5 to 13 in "**Assigning VDCP Commands**" on page 5–46
   › For PBus commands, see Steps 5 to 13 in "**Assigning PBUS Commands**" on page 5–47
   › For data modification tasks, see Steps 5 to 10 in "**Assigning Data Modification Tasks**" on page 5–48
8. In the **Add/Edit Timers** dialog, click **Commit Changes**, and then click **Done**.

## Editing a Task

You can edit any task, change the order that tasks are performed. Each time the task control is activated, all tasks on the list are performed, in the order in which they are listed.

**To edit a task**

1. Display the **Edit Component** dialog for your component.
2. Select the **Attributes** tab.
3. In the **Tasks** area, select the task you want to edit.
4. Click **Edit** to display the **Edit Task** dialog.
5. Use the provided menus to update the task.
6. Click **OK** to save your changes.
7. Click **Apply Changes** or **Apply and Close**.

**To arrange the task order:**

1. Display the **Edit Component** dialog for your component.
2. Select the **Attributes** tab.
3. Select tasks and arrange them in the order you want them to be performed. One task is performed each time the task control is activated. You can arrange tasks as follows:
   - **First** — move the selected task to the top of the task list.
   - **Move Up** — move the selected task up one on the task list.
   - **Move Down** — move the selected task down one on the task list
   - **Last** — move the selected task to the bottom of the task list.
4. Click **Apply Changes** or **Apply and Close**.

**To delete a task**

1. Display the **Edit Component** dialog for your component.
2. Select the **Attributes** tab.
3. In the **Tasks** area, select the task you want to delete.
4. Click **Delete** to remove the task from the list.
5. Click **Apply Changes** or **Apply and Close**.

# Triggering Tasks Externally

You can set up tasks to be triggered by external means such as keyboard shortcuts or RossTalk messages received from external systems. You can also use the ogScript function, **`ogscript.fireGPI.`**

Tasks are associated with certain types of components. You can create GPI triggers for these components so that when the correct message is received, the component's tasks are executed.

**Note:** External triggering is in addition to normal task execution methods. For example, if you create a GPI trigger for a push button, that button's tasks are executed when the GPI trigger message is received OR when the button is pushed.

## Creating a GPI Trigger

To create a GPI trigger, you specify a text string that, when received by the component, prompts the component to run its tasks. GPI triggers can be set only on components with task lists, such as buttons, labels, and displayed parameters.

**To create a GPI trigger:**

1. Create the component and task list.
2. In edit mode, double-click the component to open its **Edit Component** window.
3. Select the **Attributes** tab for the component type (**Param Attributes**, **Button Attributes**, or **Label Attributes**).
4. In the **GPI Trigger** box, type the trigger text.

   **Figure 5.23** shows the **GPI Trigger** box on the **Button Attributes** tab.

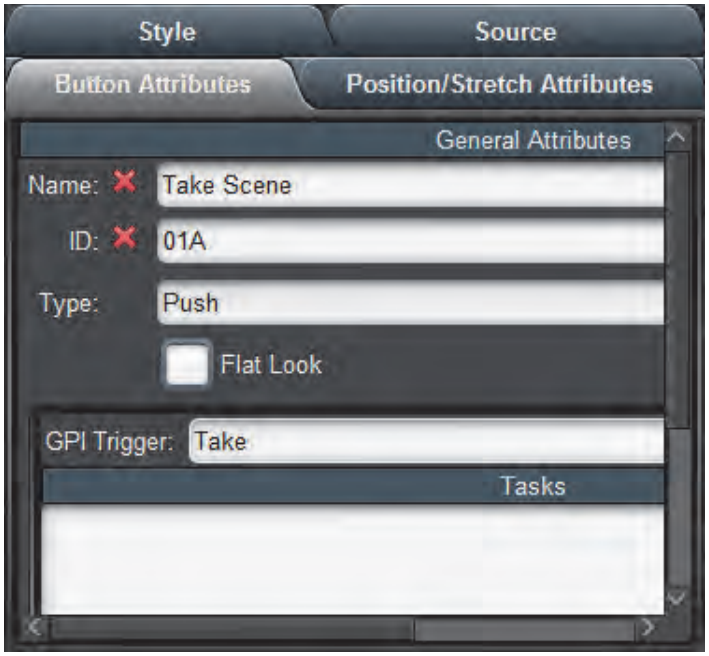



***Figure 5.23*** *- Button Attributes tab, including the GPI Trigger box.*

5. Click **Apply and Close**.

   The component is configured for external triggering of its task list.

## Triggering Tasks Using Keyboard Shortcuts

You can create a keyboard shortcut to trigger a component's task list. When panel users type the shortcut, the tasks are executed.

**To create a keyboard shortcut to trigger tasks:**

1. Create the component (button, label, or displayed parameter), complete with task list and GPI trigger.

   Note the **GPI Trigger** text.

2. On the **Window** menu, click **Preferences**.

   The **Preferences** dialog appears.

3. In the menu list on the left side of the **Preferences** dialog, click **Keyboard Shortcuts**.

   The **Keyboard Shortcuts** list appears.

4. Click the **New Command** button.

   The **Command Configuration Wizard** dialog appears.

5. In the **Available Commands** list, click **Fire GPI Trigger**.

6. Click in the **Shortcut** box, and then type the shortcut sequence.

7. In the **Active** list, select **DashBoard Context**, and then click **Next**.

8. In the **GPI Trigger** box, type the **GPI Trigger** text.

9. Optionally, if you want to send an additional piece of text data when the keyboard shortcut is used, type the data in the **State** box.

   The additional data can be used by ogScripts. For example, you may have two different keyboard shortcuts; one that triggers the task list and uses the **State** text to send an "**ON**" message to the ogScript to turn something on, and another shortcut with an "**OFF**" message.

10. In the **Execution Level** list, select one of the following:

    - **All Editors** — Sends the command to all open panels.
    - **Active Editor** — Sends the command only to the currently active panel.

11. Click **Finish**.

12. In the **Preferences** dialog, click **OK**.

## Triggering Tasks Using RossTalk Messages

A component's tasks are externally triggered when the GPI Trigger text is received by the panel. One method of sending the GPI Trigger text is to embed it in a RossTalk command. DashBoard must first be configured to listen for RossTalk commands.

The RossTalk **`sendMessage`** function is part of the ogScript language. In the following example, the GPI Trigger message "**`StartClock`**" is sent to the local computer (**`localhost`**), via port **`7788`**:

```
rosstalk.sendMessage('localhost', 7788, 'GPI StartClock');
```

You can also include state data in the message. In the following example, the same GPI trigger is sent, but with state data, **`reset`**, appended to the GPI command. You can access the state data using the **`event.getState()`** function, as shown in the second line of the example. The second line checks whether the **`event.getState()`** function exists before calling it. This is important because the function is available only when the tasks are triggered through GPI, and not when triggered by other means such as a button click or parameter change.

```
rosstalk.sendMessage('localhost', 7788, 'GPI StartClock:reset');
var resetTimer = event.getState && event.getState() == 'reset';
```

For more information about using the ogScript RossTalk object, see the ***ogScript Reference Guide (8351DR-006)***.

**To Configure DashBoard to Listen for RossTalk Commands**

1. On the **Window** menu, click **Preferences**.

   The **Preferences** dialog appears.

2. In the menu list on the left side of the **Preferences** dialog, click **Smart GPI Listener**.

   The **Smart GPI Listener** settings appear.

3. Beside **Global GPI Listener**, select **Enable**.

4. In the **Listen Port** box, specify the port for receiving RossTalk commands.

   **Tip:** The default port is **7788**.

5. Click **OK**.

## Triggering Tasks Using the ogscript.fireGPI Function

A component's tasks are externally triggered when the GPI Trigger text is received by the panel. One method of sending the GPI Trigger text is to use the **`ogscript.fireGPI`** function.

The **`ogscript.fireGPI`** function is part of the ogScript language. For more information about using ogScript, see the ***ogScript Reference Guide (8351DR-006-02)***.

# Editing Components

The **Edit Component** window enables you to configure component attributes.

**To access the Edit Component window:**

1. Select from the **Edit Mode** toolbar.

2. Double-click the component you want to edit.

   **Tip:** The components are hierarchical. As you hover over the panel, an outline appears to show which component would be selected if you were to click.

   The **Edit Component** window appears.

   The contents of the Edit Component window vary depending on the type of component you are editing.

The **Edit Component** window consists of six areas:

***Figure 5.24*** *- Edit Component WIndow*

### 1. Title Area

The title area indicates the type of component you are editing.

### 2. Insert Tag Area

This area consists of buttons that enable you to add items to the component tree. The buttons represent items which cannot be added through the graphic interface of PanelBuilder. For more information, see "**Component Tree**" on page 5–55.

### 3. Component Tree

This area shows component information for the entire CustomPanel, presented in a hierarchy. Use the component tree to select which item you want to edit. The current item is highlighted blue.

**Tip:** You can scroll through components by using the up and down keys

### 4. Attribute Editor

The attribute editor area consists of tabs listing editable attributes of the selected component. Which tabs are shown depends on the type of component you are editing.

### 5. Preview Area

This area shows the component you are editing. If you apply changes, the preview area updates.

### 6. Apply Buttons

Use the various Apply buttons to commit any changes you have made:

- **Apply Changes** — Applies the changes and keeps the Edit Component window open afterwards.
- **Apply and Close** — Applies the changes and closes the Edit Component window.
- **Close** — Closes the Edit Component window without applying any changes.

## Component Tree

This area shows component information for the entire CustomPanel, presented in a hierarchy. Use the component tree to select which item you want to edit. The current item is highlighted blue.

While most items in the tree are created by using the graphical interface of PanelBuilder, some cannot be created this way. They are created by inserting metadata tags into the tree, by clicking buttons in the **Insert Tags** area above the tree.

When you click a button, the tag is added to the tree, within a **<meta>** tag. After you add a tag, select it in the tree and then use the tabs in the **Attribute Editor** area to customize it.

You can insert the following types of tags:

- **<listener/>** — Creates a listener object, which enables DashBoard to receive messages and data from external devices over the network.

  For information about specifying listener attributes, see "**Listener Attributes Tab**" on page 5–64.
- **<style/>** — Creates a style definition. You can create named style definitions which can be referenced from anywhere. This is useful for creating common visual themes across components. By changing a setting in the style definition, you can change that setting for all components that reference that style definition.

  For information about specifying style attributes, see "**Style Tab**" on page 5–74.
- **<color/>** — Creates a color definition. You can create named color definitions which can be referenced from anywhere. This is useful for creating common visual themes across components. By changing settings in the color definition, you can change that setting for all components that reference that color definition.

  For information about specifying color attributes, see "**Color Attributes Tab**" on page 5–60.
- **<lookup/>** — Creates a data lookup table, which can be read using ogScript functions.

  For information about specifying lookup attributes, see "**Lookup Attributes Tab**" on page 5–66.
- **<ogscript/>** — Creates a named section of ogScript code, which can be referenced. You can also associate the ogScript with an action and a UI element, so that the ogScript is executed when the action is performed on the UI element. For example, you can make the ogScript run when the user drags the mouse (ondrag).

  For information about specifying ogScript attributes, see "**ogScript Attributes Tab**" on page 5–66.

## Attribute Editor Tabs

The attribute editor area of the Edit Component window consists of tabs listing editable attributes of the selected component. Which tabs are shown depends on the type of component you are editing. To edit component attributes, specify values and then apply the changes. If you want to use the default value, click the ✖ next to the attribute name.

The Source tab is present for every component. It shows the underlying XML source code that defines the selected component. You can edit the XML code directly in the Source tab, or paste XML code into it. To view or edit the XML source for the entire CustomPanel, select the top element in the component tree and then view its Source tab.

**Tip:** To quickly view the source for a component, press and hold the **Shift** key, and double-click. The **View Source** window appears. The XML source cannot be edited in this window.

Some tabs are divided into two areas: Current Attributes and Unused Attributes. The Current Attributes area shows the attribute values currently applied to the selected component. Only attributes that have been edited appear here. The Unused Attributes area shows attributes for which default values apply. For example, in the following Style tab, only the border and background attributes have been defined:

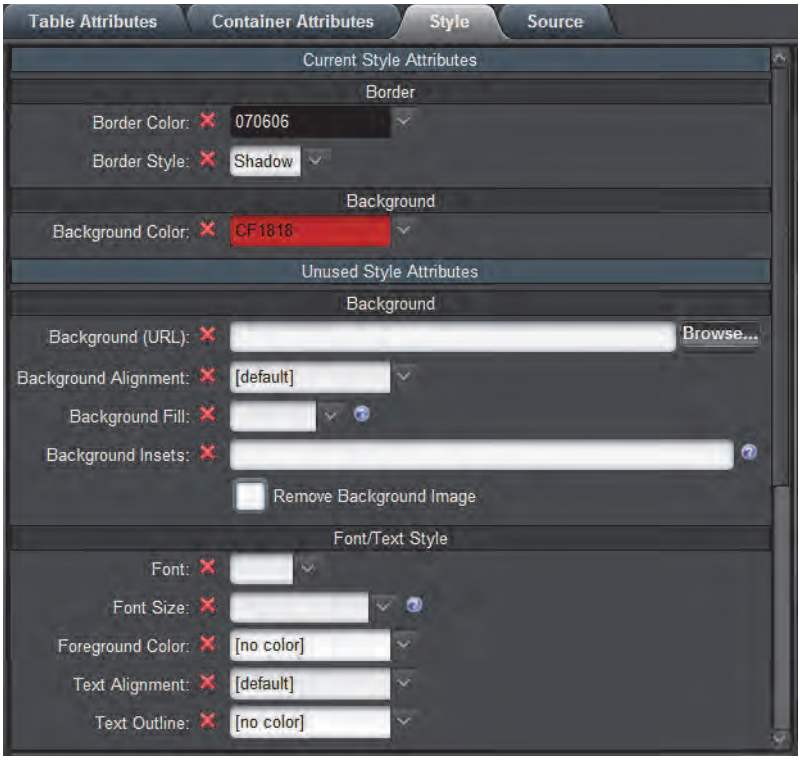



***Figure 5.25*** *- Current and Unused Attributes*

When you edit values in the Unused Attributes area and then apply the changes, those attributes move up to the Current Attributes area.

The attribute editor area includes a subset of the following tabs, depending on the type of component you are editing:

- **Abs Attributes Tab** — For more information, see "**Abs Attributes Tab**" on page 5–58.
- **Blank Attributes Tab** — For more information, see "**Blank Attributes Tab**" on page 5–59.
- **Browser Attributes Tab** — For more information, see "**Browser Attributes Tab**" on page 5–59.
- **Button Attributes Tab** — For more information, see "**Button Attributes Tab**" on page 5–60.
- **Color Attributes Tab** — For more information, see "**Color Attributes Tab**" on page 5–60.
- **Container Attributes Tab** — For more information, see "**Container Attributes Tab**" on page 5–61.
- **Dropspot Attributes Tab** — For more information, see "**Dropspot Attributes Tab**" on page 5–62.
- **Editor Attributes Tab** — For more information, see "**Editor Attributes Tab**" on page 5–62.
- **Flow Attributes Tab** — For more information, see "**Flow Attributes Tab**" on page 5–63.
- **Label Attributes Tab** — For more information, see "**Label Attributes Tab**" on page 5–64.
- **Listener Attributes Tab** — For more information, see "**Listener Attributes Tab**" on page 5–64.
- **Lookup Attributes Tab** — For more information, see "**Lookup Attributes Tab**" on page 5–66.
- **Meta Attributes Tab** — For more information, see "**Meta Attributes Tab**" on page 5–66.
- **ogScript Attributes Tab** — For more information, see "**ogScript Attributes Tab**" on page 5–66.
- **Param Attributes Tab** — For more information, see "**Param Attributes Tab**" on page 5–68.
- **Position/Stretch Attributes Tab** — For more information, see "**Position/Stretch Attributes Tab**" on page 5–70.
- **Simplegrid Attributes Tab** — For more information, see "**Simplegrid Attributes Tab**" on page 5–71.
- **Source Tab** — For more information, see "**Source Tab**" on page 5–72.
- **Split Attributes Tab** — For more information, see "**Split Attributes Tab**" on page 5–73.
- **Statuscombo Attributes Tab** — For more information, see "**Statuscombo Attributes Tab**" on page 5–73.
- **Style Tab** — For more information, see "**Style Tab**" on page 5–74.
- **Tab Attributes Tab** — For more information, see "**Tab Attributes Tab**" on page 5–77.
- **Table Attributes Tab** — For more information, see "**Table Attributes Tab**" on page 5–78.
- **Table Cell Attributes Tab** — For more information, see "**Table Cell Attributes Tab**" on page 5–79.
- **Tag Attributes Tab** — For more information, see "**Tag Attributes Tab**" on page 5–80.
- **Timer Attributes Tab** — For more information, see "**Timer Attributes Tab**" on page 5–81.
- **Timertask Attributes Tab** — For more information, see "**Timertask Attributes Tab**" on page 5–82.
- **Tr Attributes Tab** — For more information, see "**Tr Attributes Tab**" on page 5–82.
- **TreeElement Attributes Tab** — For more information, see "**Abs Attributes Tab**" on page 5–58.

## Abs Attributes Tab

Abs Attributes apply to the following types of components:

- "**Blank Canvases**" on page 5–20
- "**Image Canvases**" on page 5–22

| Attribute | Description |
|---|---|
| | **General Attributes** |
| Name | Type a name or description for the Abs canvas.<br>This name is used as a reference in the Component Tree. It does not appear in the CustomPanel interface. |
| ID | Type a scripting ID for the Abs canvas. |
| Virtual Width | Enter or select a width in pixels for the virtual width of the canvas.<br>This determines how wide the workspace canvas is. For example, if using a large monitor, you might want to use a high number of pixels to accommodate a greater number of components. Conversely, if using a smaller monitor, you might want to use a lower number of pixels.<br>Select the default check box to use the default virtual width of 1,079 pixels. |
| Virtual Height | Enter or select a height in pixels for the virtual height of the canvas.<br>This determines how tall the workspace canvas is. For example, if using a large monitor, you might want to use a high number of pixels to accommodate a greater number of components. Conversely, if using a smaller monitor, you might want to use a lower number of pixels.<br>Select the default check box to use the default virtual height of 931 pixels. |
| | **Data Source/Device Control** |
| openGear or XPression DataLinq<br><br>or<br><br>NK Series Routers | Select a device for context:<br>• **openGear or XPression DataLinq** — this option opens the **Select Device for Context** dialog box, where you can select an openGear card or XPression DataLinq XML file to associate with the panel.<br>**NK Series Routers** — this option opens the **Select IPS** dialog box, where you can select a router node.<br>**Note:** If both boxes are clear, the panel is not associated with a device or data source. |
| XPression DashBoard Linq Port | If you want to stream XML data to an XPression system directly without sharing an XML file, specify the port to use on the DashBoard computer and select **Enable Streaming**.<br>**Note:** You must also set up a DashBoard DataLinq source on the XPression DataLinq server. The DashBoard Linq must point to the IP address and port of the DashBoard computer that hosts the CustomPanel from which you want to stream data. |
| NK Series Routers | Select the check box to associate the panel with NK Series routers for device control. |

| Attribute | Description |
|---|---|
| Lock Contents (widget root) | Select this check box to disable the selection of sub-elements in the canvas. This allows for easy selection, copy, and paste of a component.<br>Elements under the widget root can only be selected by selecting the item node in the tree. Users can not add to, or directly modify, the contents of a widget, move a widget, or resize a widget. This allows the block of code for the widget to be self-contained and able to be dragged and dropped elsewhere with ease. |
| Scrolling | Click the menu and select an option for adding scroll bars to the Abs canvas:<br>• **True** — use vertical and/or horizontal scroll bar, if needed.<br>• **False** — do not use scroll bars for the Abs canvas.<br>• **Vertical** — add a vertical scroll bar to the Abs canvas.<br>• **Horizontal** — add a horizontal scroll bar to the Abs canvas.<br>• **Always** — always use vertical and horizontal scroll bars for the Abs canvas. |

## Blank Attributes Tab

Blank Attributes apply to the following types of components:

- "**Tables**" on page 5–25

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the blank component.<br>This name is used as a reference in the Component Tree. It does not appear in the CustomPanel interface. |
| ID | Type a scripting ID for the blank component. |

## Browser Attributes Tab

Browser Attributes apply to the following types of components:

- "**Web Browser Instances**" on page 5–34

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the browser component.<br>This name is used as a reference in the Component Tree. It does not appear in the CustomPanel interface. |
| ID | Type a scripting ID for the browser component. |
| URL | Type a URL of the website to use for the browser component. |

## Button Attributes Tab

Button Attributes apply to the following types of components:

- "**Buttons**" on page 5–30

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description to display on the button.<br>This name is also used as a reference in the Component Tree. |
| ID | Type a scripting ID for the button. |
| Type | Click the menu and select a button type:<br>• **Push** — select to use a simple one-press button to run tasks. The button has no state.<br>• **Toggle** — select to use a toggle button. For a toggle button, when the button is clicked, a function is turned on. Clicking it again turns the function off.<br>• **Checkbox** — select to use a check box for enabling and disabling a function.<br>• **Radio** — select to use a radio button for enabling and disabling functions. |
| Flat Look | Select the check box to use a plain, non-stylized button display. |
| **Tasks** | |
| GPI Trigger | |
| Tasks List | This list displays the tasks that have been added to a button. Tasks are commands or controls assigned to the component. The tasks run in top to bottom order. Use the buttons to arrange the tasks:<br>• **First** — click to move a selected task to the top of the **Tasks List**.<br>• **Move Up** — click to move a selected task up one position in the **Tasks List**.<br>• **Move Down** — click to move a selected task down one position in the **Tasks List**.<br>• **Last** — click to move a selected task to the bottom of the **Tasks List**. |
| Add | Click the button to open the **Add Task** dialog box and add tasks to the button. The tasks are added to the **Tasks List**. |
| Edit | Click the button to open the **Edit Task** dialog box and edit a selected task from the **Tasks List**. |
| Delete | Click the button to delete a selected task from the **Tasks List**. |

## Color Attributes Tab

Color attributes define the characteristics of a named color definition, which can be referenced in ogScript.

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name of the color definition. |
| ID | Type a scripting ID for the color definition |
| Color | Click the dropdown and select a color for the background and modify the Hue, Saturation, Lightness, and Transparency of the selected color.<br>Use the library located at the bottom of the color dropdown to select a preset color from a color swatch. |

## Container Attributes Tab

Container Attributes apply to the following types of components:

- "**Tables**" on page 5–25

| Attribute | Description |
|---|---|
| **Advanced Table Settings** | |
| Max Elements Per Row | Limits the number of table columns, for tables that are populated by a parameter that returns multiple elements, such as a set of buttons.<br>For example, this option is useful if you want to create a table of buttons, each of which includes a choice as defined in a parameter with nine values. Create a one-cell table and set Max Elements Per Row to 3. Drag the parameter onto the table, setting it as a choice list, with the Keep returned elements together option unselected. The table will have three rows of three buttons, each of which contains one of the nine choices defined in the parameter.<br>For more information on creating tables with rows of buttons, refer to the section "Buttons" on page 5-30. |
| **Position/Stretch Attributes** | |
| Anchor Points | Use the menu to specify the location and sizing of the component on the canvas should the panel be resized:<br> <br>The component remains fixed to the selected location if the panel is resized.<br>For detailed information on Anchor Points, refer to refer to the section "Anchor Points and Background Alignment" on page 5-83. |
| Top (pixels) | Type or select a number of pixels to offset the component from the top margins of the panel. |
| Left (pixels) | Type or select a number of pixels to offset the component from the left side margins of the panel. |
| Bottom (pixels) | Type or select a number of pixels to offset the component from the bottom margins of the panel. |
| Right (pixels) | Type or select a number of pixels to offset the component from the right side margins of the panel. |
| Width (pixels) | Enter or select a fixed width in pixels for the component. The component retains these dimensions if the panel is resized.<br>Select the **Use default** check box to use the default width for the component. |
| Height (pixels) | Enter or select a fixed height in pixels for the component. The component retains these dimensions if the panel is resized.<br>Select the **Use default** check box to use the default height for the component. |

## Dropspot Attributes Tab

Dropspot Attributes apply to the following types of components:

- "**Split Panels**" on page 5–23
- "**Tables**" on page 5–25

| Attribute | Description |
|---|---|
| | **General Attributes** |
| Name | Type a name or description for the dropspot.<br>This name is used as a reference in the Component Tree. It does not appear in the CustomPanel interface. |
| ID | Type a scripting ID for the dropspot. |

## Editor Attributes Tab

Editor Attributes apply to editor nodes in the component tree. Editor nodes appear in the component tree if your panel includes controls for devices that include a device editor.

| Attribute | Description |
|---|---|
| | **General Attributes** |
| Name | Type a name or description for the editor.<br>This name is used as a reference in the Component Tree. It does not appear in the CustomPanel interface. |
| ID | Type a scripting ID for the editor. |
| Lock Contents (widget root) | Select this check box to disable the selection of sub-elements in the canvas. This allows for easy selection, copy, and paste of a component.<br>Elements under the widget root can only be selected by selecting the item node in the tree. Users can not add to, or directly modify, the contents of a widget, move a widget, or resize a widget. This allows the block of code for the widget to be self-contained and able to be dragged and dropped elsewhere with ease.<br>**Note:** This option is ineffectual for embedded device editors. |
| Scrolling | Click the menu and select an option for adding scroll bars to the device canvas:<br>• **True** — use vertical and/or horizontal scroll bars according to the size of the device canvas.<br>• **False** — do not use scroll bars for the device canvas.<br>• **Vertical** — add a vertical scroll bar to the device canvas.<br>• **Horizontal** — add a horizontal scroll bar to the device canvas.<br>• **Always** — always use vertical and horizontal scroll bars for the device canvas. |

## Flow Attributes Tab

Flow Attributes apply to the following types of components:

- "**Flow Containers (Wrap Content)**" on page 5–27

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the flow container component. |
| ID | Type a scripting ID for the flow container component. |
| **Data Source/Device Control** | |
| openGear or XPression DataLinq<br><br>or<br><br>NK Series Routers | Select a device for context:<br>• **openGear or XPression DataLinq** — this option opens the **Select Device for Context** dialog box, where you can select an openGear card or XPression DataLinq XML file to associate with the flow container.<br>**NK Series Routers** — this option opens the **Select IPS** dialog box, where you can select a router node.<br>**Note:** If both boxes are clear, the flow container uses the surrounding context. |
| Horizontal Alignment | If you want the components to be neatly aligned along the right or left edge of the container, or centered within the container, specify the Horizontal Alignment accordingly. |
| Keep all widths the same | Select this option is you want the widths of all components in the container to be the same.<br>All component widths will match the width of the widest one. |
| Keep all heights the same | Select this option is you want the heights of all components in the container to be the same.<br>All component heights will match the height of the tallest one. |
| Fill single line if possible | |
| Lock Contents (widget root) | Select this check box to disable the selection of sub-elements in the canvas. This allows for easy selection, copy, and paste of a component.<br>Elements under the widget root can only be selected by selecting the item node in the tree. Users can not add to, or directly modify, the contents of a widget, move a widget, or resize a widget. This allows the block of code for the widget to be self-contained and able to be dragged and dropped elsewhere with ease. |
| Scrolling | Click the menu and select an option for adding scroll bars to the flow container component:<br>• **True** — use vertical and/or horizontal scroll bars according to the size of the flow container component.<br>• **False** — do not use scroll bars for the flow container component.<br>• **Vertical** — add a vertical scroll bar to the flow container component.<br>• **Horizontal** — add a horizontal scroll bar to the flow container component.<br>• **Always** — always use vertical and horizontal scroll bars for the flow container component. |

## Label Attributes Tab

Label Attributes apply to the following types of components:

- "**Labels**" on page 5–28
- "**Links to Device Editors or Other CustomPanels**" on page 5–29

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type the text that you want to display on the label. |
| ID | Type a scripting ID for the label. |
| Header | Select the check box to stylize the label as a header.<br>The appearance can be edited in the **Style** tab. |
| **Tasks** | |
| GPI Trigger | |
| Tasks List | This list displays the tasks that have been added to a label. Tasks are commands or controls assigned to the component. The tasks run in top to bottom order. Use the buttons to arrange the tasks:<br>• **First** — click to move a selected task to the top of the **Tasks List**.<br>• **Move Up** — click to move a selected task up one position in the **Tasks List**.<br>• **Move Down** — click to move a selected task down one position in the **Tasks List**.<br>• **Last** — click to move a selected task to the bottom of the **Tasks List**. |
| Add | Click the button to open the **Add Task** dialog box and create tasks for the label. The tasks are added to the **Tasks List**. |
| Edit | Click the button to open the **Edit Task** dialog box and edit a selected task from the **Tasks List**. |
| Delete | Click the button to delete a selected task from the **Tasks List**. |

## Listener Attributes Tab

Listener attributes define the connection information and other properties of listener objects, which enable DashBoard to receive messages and data from external devices over the network.

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the listener. |
| ID | Type a scripting ID for the listener. |
| Show start/stop toggle button | Select the check box if you want to display a button for starting and stopping the listener.<br>The button is displayed only if the <listener/> tag appears outside of the <meta/> tag in the source. Normally, <listener/> tags are used within the <meta/> tag. |
| Start automatically | Select the check box if you want the listener to start automatically.<br>**Tip:** Listeners always start automatically if the <listener> tag in the source is within a <meta> tag. |
| **Connection Type/Settings** | |

| Attribute | Description |
|---|---|
| UDP Listener | Select this option if you want the listener to detect and receive UDP messages. Specify the following:<br>• **Port** — The port number to listen to.<br>• **Max Length (UDP)** — Specify the maximum number of bytes allowed in a UDP message. The listener truncates any messages that exceed this length. |
| Listen as server | Select this option if you want the listener to be the server in the server/client relationship.<br>Specify the number of the port on which the listener will communicate. |
| Connect as client | Select this option if you want the listener to be the client in the server/client relationship.<br>Select an existing server to which the listener connects.<br>You can also create a custom connection, and add it to the Global List. The connection appears as an entry in the 'hosts' lookup. |
| Delimiter type (TCP) | If communicating over TCP, specify the delimiter type.<br>The delimiter defines the beginning of a message. |
| Delimiter value | If communicating over TCP, specify the delimiter data.<br>This available options depend on the delimiter type:<br>• **Bytes** — Specify the delimiter byte value in hexadecimal notation.<br>• **String** — Specify the delimiter string text.<br>• **Fixed Length** — Specify the message length, in bytes.<br>• **Variable Length** — Each message starts with data defining the message length. Specify how many bytes are in the length-defining data at the start of the message. |
| Sync Word (TCP) | If communicating over TCP, type the synchronization word data to be transmitted as a message header at the start of each message. |
| **Command Parsing** | |
| Stop all receive/task execution on pause | Select the check box if you want to stop receiving additional messages until all tasks associated with the previous message (including pauses) have finished execution.<br>If this option is not selected, the listener continues to fetch messages and adds them to the queue while tasks are running. This causes the task to be executed in the network RX thread.<br>If this option is selected, the task can interact directly with the socket, and possibly read additional bytes from the stream before processing continues. |
| **Tasks** | |
| Tasks List | This list displays the tasks that have been added to a listener. Tasks are commands or controls assigned to the component. The tasks run in top to bottom order.<br>Use the buttons to arrange the tasks:<br>• **First** — click to move a selected task to the top of the **Tasks List**.<br>• **Move Up** — click to move a selected task up one position in the **Tasks List**.<br>• **Move Down** — click to move a selected task down one position in the **Tasks List**.<br>• **Last** — click to move a selected task to the bottom of the **Tasks List**. |
| Add | Click the button to open the **Add Task** dialog box and add tasks to the listener. The tasks are added to the **Tasks List**. |
| Edit | Click the button to open the **Edit Task** dialog box and edit a selected task from the **Tasks List**. |
| Delete | Click the button to delete a selected task from the **Tasks List**. |

## Lookup Attributes Tab

The Lookup Attributes tab is used to edit keys and values in lookup tables, including connection entries from the Global List.

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the lookup component. |
| ID | Type a scripting ID for the lookup component. |
| Support multiline values | Select the check box to allow multiline entries in the Lookup Entries list. |
| Lookup Entries | This list displays the **Key** and **Value** that comprise entries in the lookup table.<br>Click inside a Key or Value box to edit the lookup entry. |

## Meta Attributes Tab

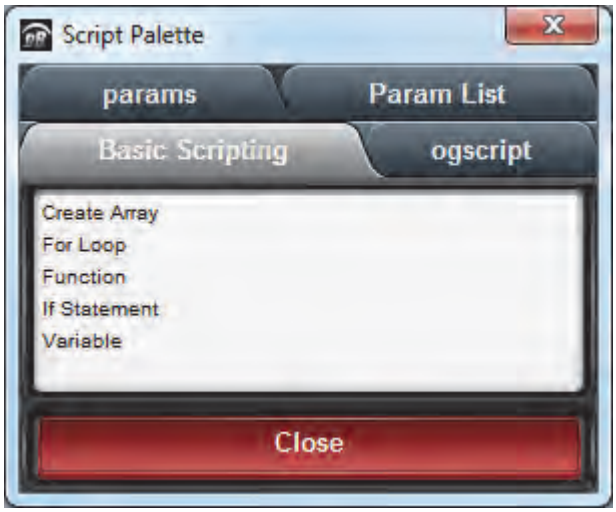
The Meta Attributes tab contains a name and OID for a metadata <meta> tag in the component tree.

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the metadata. |
| ID | Type a scripting ID for the metadata. |

## ogScript Attributes Tab

The ogScript Attributes tab enables you to set the properties of named sections of ogScript code, which can be referenced.

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the section of ogScript. |
| ID | Type a scripting ID for the section of ogScript. |
| Event Type(s) | If you want the ogScript to execute when a specific event occurs in relation to a certain UI element, select the event type from the list. You can select multiple event types. |
| Target | If you specified one or more event types for triggering execution, select the ID of the corresponding UI element. The ogScript is executed when one of the selected events occurs in relation to the selected target. |
| **ogScript Content** | |

| Attribute | Description |
|---|---|
| Script Palette | Select this check box if you want the script palette to be available as you create your ogScript.<br>The script palette is a tool that enables you to quickly add ogScript command templates, which you can customize to produce code more quickly and with fewer coding errors.<br>Templates enable you to more easily create ogScript functions, manipulate parameters, etc.<br>  <br><br>***Figure 5.26*** *- The Script Palette*<br>**To use the script palette:**<br>• In the source editor area, click the position where you want to insert a command template.<br>• In the script palette, click the tab that contains the command template you want to use (params, Basic Scripting, ogscript, Param List).<br>• If you are using the **Param List**, select whether you want to **Get** a value or **Set** a value.<br>• Select the command (or parameter) and drag it into the script window.The template is inserted.<br>• Edit the code, replacing the colored placeholders as required. |
| Search | If you want to find a particular string within your code, type the string in the **Search** box, and click **Find Next**.<br>**Tips:**<br>• To immediately find the same string again, press **Enter**.<br>• To move the cursor into the **Search** box, press **Ctrl+f**. |
| Scripting Area | The blank area with line numbers along the left edge, where you type or paste the ogScript code. |

## Param Attributes Tab

The Param Attributes tab is used to configure the attributes of any item on the panel that contains a value, whether it is from a device or is a user-created parameter.

This is for both user-created items and for items that have been dragged and dropped into the canvas.

| Attribute | Description |
|---|---|
| **Parameter Settings** | |
| Name (read-only) | The parameter selected for the CustomPanel. |
| OID (Object Identifier) | Select a parameter from the menu to display its information in the tab and to edit its properties. The currently available parameters are listed according to your current data source. Their assigned OID tags and are available for the selected CustomPanel either from:<br>• an associated data source<br>• as defined via one of the tools in the **Edit Mode** toolbar.<br>• a component dragged and dropped from a device in the Tree View.<br>Users will almost never use this field to modify the OID or change the parameter they are controlling. The more useful method is to drag new ones, delete old ones, etc. |
| Menu | This is a menu identifier. |
| Constraint | Use the menu to set limitations on the parameter values. These options depend on what has been selected as the **Type** in the Add/Edit Parameters dialog box.<br>By default, these values are determined directly from the parameter that is referenced. However, these defaults can be overridden by selecting the **Override constraint** check box. Overriding the defaults does not modify the constraints for the parameter, only for this particular control in the custom panel.<br>The possible options are as follows:<br>• **Unconstrained** — select this when using a string, integer, or float type. No limitations are applied to the parameter value. For example, a text field parameter where a user can type any word or mix of letters and numbers.<br>• **Range Constraint** — select this when using an integer or float type. Use this option to stipulate a range of numbers that the user can select from, such as a minimum and maximum value.<br>• **Choice Constraint** — select this to provide a specific list of options that the user chooses from where each option is associated with a tag.<br>• **Alarm Table** — select this to set constraint values for alarm states.<br>• **String Choice** — select this when using a string type to provide a specific list of strings from which the user chooses.<br>• The constraint values are located in the **Constraint Value** field. |
| Precision | Type or select a value:<br>• When used with numbers — This field defines the number of digits following the decimal point displayed for printed numbers. It applies mainly to floating point numbers.<br>• When used with string arrays — This field defines the maximum number of bytes reserved for a single element in the array. If it is 0, the maximum number of bytes in a parameter value are shared arbitrarily amongst all elements in the array. |

| Attribute | Description |
|---|---|
| Constraint Value | Use the **Constraint Value** area to define the valid set of values for the parameter.<br><br>For choice constraints, including string choice, do the following once for each valid value:<br>• In the **Value** column, click **[insert value]**, type a valid value, and then press **Enter**.<br>• In the **Name** column, type a name for the value. The name is associated with the parameter value, and appears on labels, etc. The **Name** column is available only if the parameter is a numeric type.<br><br>For range constraints:<br>• In the **Minimum** box, type the lowest valid value.<br>• In the **Maximum** box, type the highest valid value.<br>• In the **Step Size** column, type the step size. For example, if valid values must be evenly divisible by 10, type 10.<br>• If you plan to use a touch wheel in your panel, select the **Loop** check box.<br><br>For alarm table constraints, do the following once for each valid value:<br>• In the **Bit** box, type the bit value for the constraint value. For example, you may have two options; 1 and 0. You would have one row for each bit state. The bit must be unique for each constraint value.<br>• In the **Severity** box, select a severity level.<br>• In the **String** box, type the alarm message you want associated with this constraint value. |
| Widget Hint | Select the graphical display hint for the parameter using the menu. The options are as follows:<br>• **Default** — displays the parameter as defined according to the data source.<br>• **Read-only text** — displays the parameter as a status text field that can not be altered by the user. A border and background are automatically applied to the field.<br>• **Label** — displays the parameter as a text field without a border or background.<br>• **Text Entry** — displays the parameter as a single line text field that is editable by the user. The user must enter one of the values defined using the **Constraint Value** field.<br>• **Multi-Line Text Entry** — displays the parameter as a text field with more than one line. The user must enter one of the values defined using the **Constraint Value** field.<br>• **HTML Content** — displays the parameter as a field that requires the user to input HTML data.<br>• **Editable Dropdown List** — displays the parameter as a menu that the user clicks to display an expanded list of values to choose from. These values are determined in the **Constraint Value** field.<br>• **Alarm-Style Colored Dot** — displays the parameter as a status indicator, similar to an LED, that updates based on conditions defined in the **Constraint Value** field.<br>Select the **Get value from parameter box** to disable the **Widget Hint** menu and use a graphical display hint from the parameter. |
| Force Read Only | Select this check box to use the parameter as read-only. |
| Keep returned elements together | Select the check box to keep parameters from devices together in the layout of radio buttons, toggle buttons, etc. |
| Current Value (read-only) | Displays the device that the parameter is used for. |
| **Tasks** | |

| Attribute | Description |
|---|---|
| GPI Trigger | |
| Tasks List | This list displays the tasks that have been associated with the parameter. When the parameter value changes, the tasks are performed in the order listed. Use the buttons to arrange the order:<br>• **First** — click to move a selected task to the top of the **Tasks List**.<br>• **Move Up** — click to move a selected task up one position in the **Tasks List**.<br>• **Move Down** — click to move a selected task down one position in the **Tasks List**.<br>• **Last** — click to move a selected task to the bottom of the **Tasks List**. |
| Add | Click the button to open the **Add Task** dialog box and create tasks for the parameter. The tasks are added to the **Tasks List**. |
| Edit | Click the button to open the **Edit Task** dialog box and edit a selected task from the **Tasks List**. |
| Delete | Click the button to delete a selected task from the **Tasks List**. |
| **General Attributes** | |
| Name | Type a name or description for the parameter. |
| ID | Type a scripting ID for the parameter.<br>Scripting IDs are helpful because they provide a name for the parameter in the tree. |

## Position/Stretch Attributes Tab

Position/Stretch Attributes apply to the following types of components:

- "**Blank Canvases**" on page 5–20
- "**Tab Groups**" on page 5–21
- "**Image Canvases**" on page 5–22
- "**Labels**" on page 5–28
- "**Links to Device Editors or Other CustomPanels**" on page 5–29
- "**Buttons**" on page 5–30
- "**Web Browser Instances**" on page 5–34

| Attribute | Description |
|---|---|
| **Position/Stretch Attributes** | |
| Anchor Points | Use the menu to specify the location and sizing of the component on the canvas should the panel be resized:<br> <br>The component remains fixed to the selected location if the panel is resized.<br>For detailed information on Anchor Points, refer to refer to the section "Anchor Points and Background Alignment" on page 5-83. |
| Top (pixels) | Type or select a number of pixels to offset the component from the top margins of the panel. |
| Left (pixels) | Type or select a number of pixels to offset the component from the left side margins of the panel. |
| Bottom (pixels) | Type or select a number of pixels to offset the component from the bottom margins of the panel. |

| Attribute | Description |
|---|---|
| Right (pixels) | Type or select a number of pixels to offset the component from the right side margins of the panel. |
| Width (pixels) | Enter or select a fixed width in pixels for the component. The component retains these dimensions if the panel is resized.<br>Select the **Use default** check box to use the default width for the component. |
| Height (pixels) | Enter or select a fixed height in pixels for the component. The component retains these dimensions if the panel is resized.<br>Select the **Use default** check box to use the default height for the component. |

## Simplegrid Attributes Tab

Simplegrid Attributes apply to the following types of components:

- "**Simple Grids**" on page 5–26

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the simplegrid component. |
| ID | Type a scripting ID for the simplegrid component. |
| **Data Source/Device Control** | |
| openGear or XPression DataLinq<br><br>or<br><br>NK Series Routers | Select a device for context:<br>• **openGear or XPression DataLinq** — this option opens the **Select Device for Context** dialog box, where you can select an openGear card or XPression DataLinq XML file to associate with the simplegrid.<br>**NK Series Routers** — this option opens the **Select IPS** dialog box, where you can select a router node.<br>**Note:** If both boxes are clear, the table uses the surrounding context. |
| Rows | If you want to specify the maximum number of rows in the simple grid, select **Override Default** and then specify the number of rows.<br>By default there is only one row. The grid is divided into as many columns as the number of components you insert. |
| Columns | If you want to specify the number of columns in the simple grid, select **Override Default** and then specify the number of columns.<br>**Note:** If the simple grid contains more components than there are cells to hold them, additional columns are created. |

| Attribute | Description |
|---|---|
| Lock Contents (widget root) | Select this check box to disable the selection of sub-elements in the canvas. This allows for easy selection, copy, and paste of a component.<br>Elements under the widget root can only be selected by selecting the item node in the tree. Users can not add to, or directly modify, the contents of a widget, move a widget, or resize a widget. This allows the block of code for the widget to be self-contained and able to be dragged and dropped elsewhere with ease. |
| Scrolling | Click the menu and select an option for adding scroll bars to the simplegrid component:<br>• **True** — use vertical and/or horizontal scroll bars according to the size of the simplegrid component.<br>• **False** — do not use scroll bars for the simplegrid component.<br>• **Vertical** — add a vertical scroll bar to the simplegrid component.<br>• **Horizontal** — add a horizontal scroll bar to the simplegrid component.<br>• **Always** — always use vertical and horizontal scroll bars for the simplegrid component. |

## Source Tab

In the Edit Component window, the Source tab is present for every component. It shows the underlying XML source code that defines the selected component. You can edit the XML code directly in the Source tab, or paste XML code into it. To view or edit the XML source for the entire CustomPanel, select the top element in the component tree and then view its Source tab.

**Tip:** To quickly view the source for a component, press and hold the **Shift** key, and double-click. The **View Source** window appears. The XML source cannot be edited in this window.

| Editor Feature | Description |
|---|---|
| Line Wrap | Select the check box to enable line wrapping, so all the code is visible without scrolling horizontally. |
| ogScript Code | When selected, the code display is optimized for editing ogScript.<br>When cleared, the code display is optimized for editing OGLML (XML). |
| Search | If you want to find a particular string within your code, type the string in the **Search** box, and click **Find Next**.<br>**Tips:**<br>• To immediately find the same string again, press **Enter**.<br>• To move the cursor into the **Search** box, press **Ctrl+f**. |

## Split Attributes Tab

Split Attributes apply to the following types of components:

- "**Split Panels**" on page 5–23

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the split panel component. |
| ID | Type a scripting ID for the split panel component. |
| **Data Source/Device Control** | |
| openGear or XPression DataLinq<br><br>or<br><br>NK Series Routers | Select a device for context:<br>• **openGear or XPression DataLinq** — this option opens the **Select Device for Context** dialog box, where you can select an openGear card or XPression DataLinq XML file to associate with the split panel.<br>**NK Series Routers** — this option opens the **Select IPS** dialog box, where you can select a router node.<br>**Note:** If both boxes are clear, the split panel uses the surrounding context. |
| XPression DashBoard Linq Port | If you want to stream XML data to an XPression system directly without sharing an XML file, specify the port to use on the DashBoard computer, and select **Enable Streaming**.<br>**Note:** You must also set up a DashBoard DataLinq source on the XPression DataLinq server. The DashBoard Linq must point to the IP address and port of the DashBoard computer that hosts the CustomPanel from which you want to stream data. |
| NK Series Routers | Select the check box to associate the component with NK Series routers for device control. |
| Lock Contents (widget root) | Select this check box to disable the selection of sub-elements in the canvas. This allows for easy selection, copy, and paste of a component.<br>Elements under the widget root can only be selected by selecting the item node in the tree. Users can not add to, or directly modify, the contents of a widget, move a widget, or resize a widget. This allows the block of code for the widget to be self-contained and able to be dragged and dropped elsewhere with ease. |
| Scrolling | Click the menu and select an option for adding scroll bars to the split panel:<br>• **True** — use vertical and/or horizontal scroll bars according to the size of the split panel.<br>• **False** — do not use scroll bars for the split panel.<br>• **Vertical** — add a vertical scroll bar to the split panel.<br>• **Horizontal** — add a horizontal scroll bar to the split panel.<br>• **Always** — always use vertical and horizontal scroll bars for the split panel. |

## Statuscombo Attributes Tab

Statuscombo Attributes apply to devices in the component tree with status dot indicators. Statuscombo nodes appear in the component tree if your panel includes devices that include a status indicator.

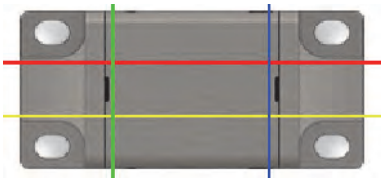
| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the device. |
| ID | Type a scripting ID for the device. |

## Style Tab

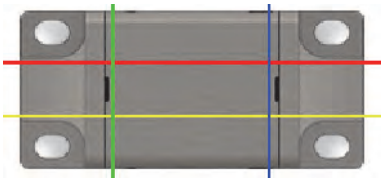
In the Edit Component window, the Style tab is present for every component.

If you are setting the style for a checkbox button or toggle button, you can specify separate styles for each button state. The Style tab has a subtab for each state (Default, Toggle On, Toggle Off).

| Attribute | Description |
|---|---|
| | **Background** |
| Background (URL) | Type a URL or file path to an image to use as the background for the component, or click **Browse** and use the **Open** dialog box to locate a file. |
| Background Alignment | Use the menu to specify where to position the background of the component in relation to the component face:<br> <br>The component remains fixed to the selected location if the panel is resized.<br>For detailed information on Anchor Points and Background Alignment, refer to the section “Anchor Points and Background Alignment” on page 5-83. |
| Background Color | Click the dropdown and select a color for the background and modify the Hue, Saturation, Lightness, and Transparency of the selected color.<br>Use the library located at the bottom of the color dropdown to select a preset color from a color swatch. |
| Background Fill | Click the menu and select a fill mode for the background:<br>• **Crop** — crop the image to fit the canvas.<br>• **None** — remove the image from the canvas.<br>• **Horizontal** — stretch the image horizontally on the canvas according to the **Background Alignment** menu. For example, if the **Background Alignment** is set to **Center**, the image will be stretched horizontally in the canvas and centered on the canvas.<br>• **Vertical** — stretch the image vertically on the canvas according to the **Background Alignment** menu. For example, if the **Background Alignment** is set to **Center**, the image will be stretched vertically in the canvas and centered on the canvas.<br>• **Shrink** — scale the image to fit the canvas area.<br>• **Fit** — fit the image to the canvas.<br>• **Tile** — repeat the image in a grid pattern on the canvas.<br>• **Both** — resize the image to fit the canvas.<br>• **Paint9** — position the image according to nine-box layout within the canvas. |

| Attribute | Description |
|---|---|
| Background Insets | Type a number in pixels for the background inset.<br>Insets are groups of four numbers (in pixels): top, left, bottom, right. The Background Insets are used to define the corners of a 9-box background. A 9-box is when the corners of the box are a fixed size, the sides have a fixed width, the top and bottom have a fixed height, and the center grows.<br>For example, for the following image:<br>  <br>• top = red<br>• left = green<br>• bottom = yellow<br>• right = blue<br>The image is 342 x 155 pixels. The left is 95 pixels and the top is 50 pixels. |
| Remove Background Image | Select the check box to remove the background image from the component. |
| **Border** | |
| Border Color | Click the dropdown and select a color for the border and modify the Hue, Saturation, Lightness, and Transparency of the selected color.<br>Use the library located at the bottom of the color dropdown to select a preset color from a color swatch. |
| Border Style | Click the menu and select a style for the canvas border:<br>• **Shadow** — use a dropdown shadow of the edges of the canvas as the border.<br>• **None** — do not use a border.<br>• **Etched** — use an engraved appearance as the border for the edges of the canvas.<br>The border style is mutually exclusive with color. |
| **Font/Text Style** | |
| Font | Click the menu and select a font:<br>• **Mono** — use a typewriter style font.<br>• **Default** — use the default font. The default font is Arial Bold.<br>• **Bold** — use a bold font. |
| Font Size | Click the menu and select a font size for the text ranging from **Smallest** to **Biggest**, or enter a font point size in the box.<br>The text will be truncated if it does not fit on the canvas area. |
| Foreground Color | Click the dropdown and select a color for the text and modify the Hue, Saturation, Lightness, and Transparency of the selected color.<br>Use the library located at the bottom of the color dropdown to select a preset color from a color swatch. |

| Attribute | Description |
|---|---|
| Text Alignment | Click the menu and select a position for the alignment of the text on the component canvas:<br> <br>• select the top option to position the text at the top of the component canvas.<br>• select the left option to position the text to the left side of the component canvas.<br>• select the center option to position the text in the center of the component canvas.<br>• select the right option to position the text to the right side of the component canvas.<br>• select the bottom option to position the text at the bottom of the component canvas. |
| Text Outline | Click the dropdown and select a color for the border of the text and modify the Hue, Saturation, Lightness, and Transparency of the selected color.<br>Use the library located at the bottom of the color dropdown to select a preset color from a color swatch. |
| **Icons** | |
| Drag Icon (URL) | Type or select the URL of a web page or a file to open when the user drags this icon to the editor area or selects the node and clicks **Open**.<br>Select the check box to remove the drag icon. |
| Hover Icon (URL) | Type or select the URL of a web page or a file to open when the user hovers over this icon or selects the node and clicks **Open**.<br>The Hover Icon is only applicable to buttons.<br>Select the check box to remove the hover icon. |
| Icon (URL) | Type or select the URL of an image to use as the icon for the node in the DashBoard tree view.<br>Select the check box to remove the icon. |
| **Other** | |
| Defined Style | Click the menu and select a pre-configured style for the component.<br>You can create your own predefined styles in OGLML by adding a `<style id=""/>` tag. Predefined styles can be overridden. |

| Attribute | Description |
|---|---|
| Insets | Type an inset number in pixels for the defined style.<br>Insets are groups of four numbers (in pixels): top, left, bottom, right. The Insets are used to define the corners of a 9-box background. A 9-box is when the corners of the box are a fixed size, the sides have a fixed width, the top and bottom have a fixed height, and the center grows. They are used to define the margins around the text/icon of a button widget.<br>For example, for the following image:<br>  <br>• top = red<br>• left = green<br>• bottom = yellow<br>• right = blue<br>The image is 342 x 155 pixels. The left is 95 pixels and the top is 50 pixels. |
| Tooltip | Type a tip or any relevant information to display when the cursor is hovered over the component. |

## Tab Attributes Tab

Tab Attributes apply to the following types of components:

- "**Tab Groups**" on page 5–21

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the tab component. |
| ID | Type a scripting ID for the tab component. |
| **Data Source/Device Control** | |
| openGear or XPression DataLinq<br><br>or<br><br>NK Series Routers | Select a device for context:<br>• **openGear or XPression DataLinq** — this option opens the **Select Device for Context** dialog box, where you can select an openGear card or XPression DataLinq XML file to associate with the tab.<br>**NK Series Routers** — this option opens the **Select IPS** dialog box, where you can select a router node.<br>**Note:** If both boxes are clear, the table uses the surrounding context. |
| XPression DashBoard Linq Port | If you want to stream XML data to an XPression system directly without sharing an XML file, specify the port to use on the DashBoard computer, and select **Enable Streaming**.<br>**Note:** You must also set up a DashBoard DataLinq source on the XPression DataLinq server. The DashBoard Linq must point to the IP address and port of the DashBoard computer that hosts the CustomPanel from which you want to stream data. |
| **Advanced Tab Settings** | |
| Tab Placement | Click the menu and select a position for the placement of the tabs on the canvas.<br>The tabs are placed on top of each other without any visual tabs, allowing you to create links to each tab and flip between them. |

| Attribute | Description |
|---|---|
| Lock Contents (widget root) | Select this check box to disable the selection of sub-elements in the canvas. This allows for easy selection, copy, and paste of a component.<br>Elements under the widget root can only be selected by selecting the item node in the tree. Users can not add to, or directly modify, the contents of a widget, move a widget, or resize a widget. This allows the block of code for the widget to be self-contained and able to be dragged and dropped elsewhere with ease. |
| Scrolling | Click the menu and select an option for adding scroll bars to the tab component:<br>• **True** — use vertical and/or horizontal scroll bars according to the size of the tab component.<br>• **False** — do not use scroll bars for the tab component.<br>• **Vertical** — add a vertical scroll bar to the tab component.<br>• **Horizontal** — add a horizontal scroll bar to the tab component.<br>• **Always** — always use vertical and horizontal scroll bars for the tab component. |

## Table Attributes Tab

Table Attributes apply to the following types of components:

- "**Tables**" on page 5–25

| Attribute | Description |
|---|---|
| | **General Attributes** |
| Name | Type a name or description for the table component. |
| ID | Type a scripting ID for the table component. |
| | **Data Source/Device Control** |
| openGear or XPression DataLinq<br><br>or<br><br>NK Series Routers | Select a device for context:<br>• **openGear or XPression DataLinq** — this option opens the **Select Device for Context** dialog box, where you can select an openGear card or XPression DataLinq XML file to associate with the table.<br>**NK Series Routers** — this option opens the **Select IPS** dialog box, where you can select a router node.<br>**Note:** If both boxes are clear, the table uses the surrounding context. |
| XPression DashBoard Linq Port | If you want to stream XML data to an XPression system directly without sharing an XML file, specify the port to use on the DashBoard computer, and select **Enable Streaming**.<br>**Note:** You must also set up a DashBoard DataLinq source on the XPression DataLinq server. The DashBoard Linq must point to the IP address and port of the DashBoard computer that hosts the CustomPanel from which you want to stream data. |

| Attribute | Description |
|---|---|
| Lock Contents (widget root) | Select this check box to disable the selection of sub-elements in the canvas. This allows for easy selection, copy, and paste of a component.<br>Elements under the widget root can only be selected by selecting the item node in the tree. Users can not add to, or directly modify, the contents of a widget, move a widget, or resize a widget. This allows the block of code for the widget to be self-contained and able to be dragged and dropped elsewhere with ease. |
| Scrolling | Click the menu and select an option for adding scroll bars to the table component:<br>• **True** — use vertical and/or horizontal scroll bars according to the size of the table component.<br>• **False** — do not use scroll bars for the table component.<br>• **Vertical** — add a vertical scroll bar to the table component.<br>• **Horizontal** — add a horizontal scroll bar to the table component.<br>• **Always** — always use vertical and horizontal scroll bars for the table component. |

## Table Cell Attributes Tab

Table Cell Attributes apply to the following types of components:

- "**Tables**" on page 5–25

Table cell attributes can also apply to buttons if a table of buttons is created:

- "**Buttons**" on page 5–30

| Attribute | Description |
|---|---|
| | **Table Cell Alignment** |
| Alignment | If a table cell is not set to fill both and the control is smaller than the cell, this controls where the cell's control is anchored in the cell. Use the menu to specify where to position the table cell:<br> <br>The table cell remains fixed to the selected location if the table is resized. |
| Fill | Click the menu and select a fill mode for the table cell:<br>• **None** — remove the image from the table cell.<br>• **Horizontal** — stretch the image horizontally in the table cell according to the **Alignment** menu. For example, if the **Alignment** is set to **Center**, the image will be stretched horizontally in the table cell and centered in the table cell.<br>• **Vertical** — stretch the image vertically in the table cell according to the **Alignment** menu. For example, if the **Alignment** is set to **Center**, the image will be stretched vertically in the table cell and centered in the table cell.<br>• **Both** — resize the image to fit the table cell. |
| Horizontal Weight (%) | Use the slide bar to determine the amount of horizontal space in the column for the table cell. The size of a cell is calculated by comparing the relative values of the cells across a column.<br>DashBoard will use the largest value in the column. If you want to reduce the column size, the weight of all cells in the column need to be shrunk.<br>Users might find it convenient to have the values across a column add to 100%. |

| Attribute | Description |
|---|---|
| Vertical Weight (%) | Use the slide bar to determine the amount of vertical space in the row for the table cell.The size of a cell is calculated by comparing the relative values of the cells across a row.<br>DashBoard will use the largest value in the row. If you want to reduce the row size, the weight of all cells in the row need to be shrunk.<br>Users might find it convenient to have the values across a row add to 100%. |
| Colspan | Type or select an amount of column cells for the cell to span.<br>Select the **Default** check box to use the default column span. |
| Rowspan | Type or select an amount of row cells for the cell to span.<br>Select the **Default** check box to use the default row span. |
| Width | Type or select an amount of pixels for the width of the cell.<br>Select the **Default** check box to use the default cell width. The default is 1. |
| Height | Type or select an amount of pixels for the height of the cell.<br>Select the **Default** check box to use the default cell height. The default is 1. |
| Insets | Type an inset number for the table cell<br>Insets are groups of four numbers: top, left, bottom, right. They are used to define the margins around the text/icon of a button widget. |
| **Advanced Attributes** | |
| Orientation | If multiple elements are returned, the selected orientation determines if the elements are placed in rows (vertical) or columns (horizontal). Click the menu and select an orientation:<br>• **Horizontal** — select to orientate the table cell horizontally. Use the **Expected # Elements** to configure the number of elements to span.<br>• **Vertical** — select to orientate the table cell vertically. Use the **Expected # Elements** to configure the number of elements to span. |
| Expected # Elements | Enter an amount of elements to fill the table if multiple elements are returned. If a small number of elements, or none, are returned, this number of elements will create empty cells in order to maintain the table structure. |

## Tag Attributes Tab

Tag Attributes apply to the following types of components:

- "**Blank Canvases**" on page 5–20
- "**Tab Groups**" on page 5–21
- "**Image Canvases**" on page 5–22
- "**Split Panels**" on page 5–23
- "**Tables**" on page 5–25
- "**Labels**" on page 5–28
- "**Links to Device Editors or Other CustomPanels**" on page 5–29
- "**Buttons**" on page 5–30
- "**Web Browser Instances**" on page 5–34

| Attribute | Description |
|---|---|
| **External Link** | |
| All Connections | Select the radio button to specify a device on the network for the link.<br>Use the menu to select a device from the **Basic Tree View**. |

| Attribute | Description |
|---|---|
| File Navigator | Select the radio button to specify a *.grid file for the link.<br>Use the menu to select a file from the **File Navigator**. |
| Local File | Select the radio button to specify a file from a local computer or a server for the link.<br>Click **Browse** to use the **Open** dialog box to select a file from a location on a local computer or sever. |
| Button Style | Select a method for displaying the link for the component:<br>• **Button** — select to use a button component as the style for the link.<br>• **Label** — select to use a label component as the style for the link. |
| **General Attributes** | |
| Name | Type a name or description for the link. |
| ID | Type a scripting ID for the link. |

## Timer Attributes Tab

The Timer Attributes apply to Timer widgets:

- "**Timers**" on page 5–40

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the timer. |
| ID | Type a scripting ID for the timer. The default is Timer ID. |
| **Timer Properties** | |
| Timer ID | Type a name for the timer. |
| Display | Specify the time format for the timer:<br>• Use the list to select the format.<br>• Type the format and then press **Enter**.<br>To view descriptions of the time formatting symbols, click beside the **Display** list. |
| Time Source | Select a radio button to specify the type of timer to use:<br>• **Self** — starts and stops manually, and is independent of any other timer.<br>• **Simple clock** — matches the time clock of the local computer.<br>• **Time Until** — counts down the amount of time remaining until a future date and time.<br>• **Other Timer** — links this timer to another timer. Starting or stopping the linked timer also starts or stops this timer. |
| Start | Specify the timer value to begin counting from. Only **Self** timers use this value. |
| Stop | Specify the timer value to stop counting at. Only **Self** and **Time Until** timers use this value. |
| Repeat Rate | In the **Every** box, enter or select an amount of time to control how often the timer tasks are run.<br>Use the list next to the **Every** box to select a unit of time for the repeat rate.<br>For more information about the repeat rate, click beside the list used to select a unit of time. |
| **Tasks** | |

| Attribute | Description |
|---|---|
| Tasks List | This list displays the tasks that have been added to a label. Tasks are commands or controls assigned to the component. Use the buttons to arrange the tasks:<br>• **First** — click to move a selected task to the top of the **Tasks List**.<br>• **Move Up** — click to move a selected task up one position in the **Tasks List**.<br>• **Move Down** — click to move a selected task down one position in the **Tasks List**.<br>• **Last** — click to move a selected task to the bottom of the **Tasks List**. |
| Add | Click the button to open the **Add Task** dialog box and create tasks for the timer. The tasks are added to the **Tasks List**. |
| Edit | Click the button to open the **Edit Task** dialog box and edit a selected task from the **Tasks List**. |
| Delete | Click the button to delete a selected task from the **Tasks List**. |

## Timertask Attributes Tab

Timertask Attributes apply to the following types of components:

- "**Timers**" on page 5–40

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the timer task. |
| ID | Type a scripting ID for the timer task. |

## Tr Attributes Tab

TableRow (Tr) Attributes apply to the following types of components:

- "**Tables**" on page 5–25

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the table row. |
| ID | Type a scripting ID for the table row. |

## TreeElement Attributes Tab

TreeElement Attributes apply to Statuscombo attributes.

| Attribute | Description |
|---|---|
| **General Attributes** | |
| Name | Type a name or description for the link. |
| ID | Type a scripting ID for the link. |

## Anchor Points and Background Alignment

Anchor Points and Background Alignment determine how an object moves if the user interface is resized for different monitor and window sizes. They are relative to the container in which they are located (for example, a tab, a split pane, etc.). In the Component Tree, they are at very least in the top level of the canvas.

Anchor Points or Background Alignment can be applied in three different ways:

- Anchoring to a corner of the canvas

Select the top left, top right, bottom left, or bottom right quadrant from the Anchor Points/Background Alignment menu to anchor the component to a corner of the canvas represented by the selected corner quadrant in the menu. When the canvas is expanded or shrunk, the component will remain a fixed distance from the corner edges of the canvas according to the set pixel amounts for the selected quadrant in the menu:

› **top** and **left** for the top-left quadrant

The component will be anchored to the top-left of the canvas and use a fixed width and height.

› **top** and **right** for the top-right quadrant

The component will be anchored to the top-right of the canvas and use a fixed width and height.

› **left** and **bottom** for the bottom-left quadrant

The component will be anchored to the bottom-left of the canvas and use a fixed width and height.

› **bottom** and **right** for the bottom-right quadrant

The component will be anchored to the bottom-right of the canvas and use a fixed width and height.

For example, the blank canvas component in **Figure 5.27** is anchored to the top left corner of the canvas.

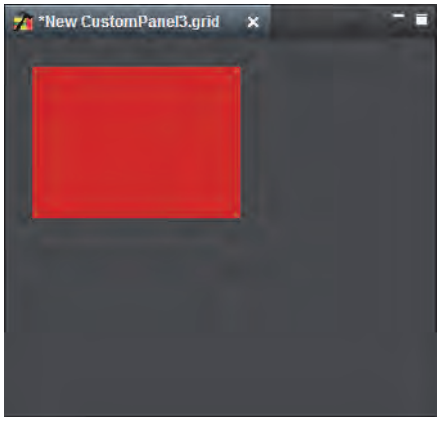



***Figure 5.27*** *- Component anchored to corner of panel*

If the panel is resized, the blank canvas component remains in the same position from the top left edges of the panel and remains the same dimensions as the panel is expanded, as demonstrated in **Figure 5.28**.

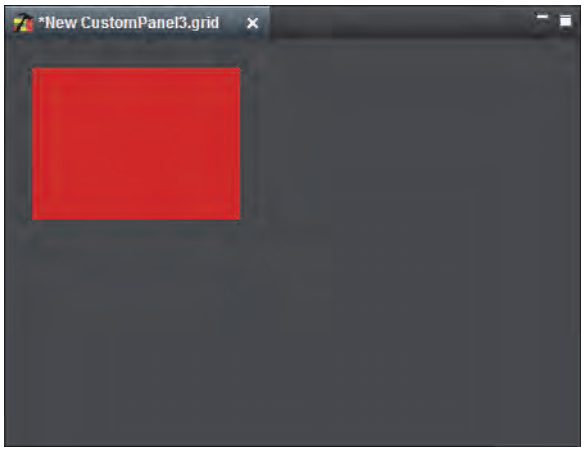



***Figure 5.28*** *- Component anchored to corner while panel is expanded*

- Anchoring to an edge of the canvas

Select the top, right, bottom, or left quadrant from the Anchor Points/Background Alignment menu to anchor the component to an edge of the canvas represented by the selected edge quadrant in the menu. Depending on the selected quadrant, when the canvas is expanded or shrunk, the component will remain a fixed distance from the top, right, bottom, or left edges of the canvas according to the set pixel amounts, and will expand or shrink:

› **top**, **left**, and **right** for the top quadrant

The component will be anchored to the top side of the canvas. Adjust the component horizontally with a fixed offset from the left and right edges of the canvas while using a fixed a height.

› **top**, **bottom**, and **right** for the right quadrant

The component will be anchored to the right side of the canvas. Adjust the component vertically with a fixed offset from the top and bottom edges of the canvas while using a fixed a width.

› **left**, **bottom**, and **right** for the bottom edge quadrant

The component will be anchored to the bottom side of the canvas. Adjust the component horizontally with a fixed offset from the left and right edges of the canvas while using a fixed a height.

› **top**, **left**, and **bottom** for the bottom right quadrant

The component will be anchored to the left side of the canvas. Adjust the component vertically with a fixed offset from the top and bottom edges of the canvas while using a fixed a width.

For example, the blank canvas component in **Figure 5.29** is anchored to the edge of the canvas.

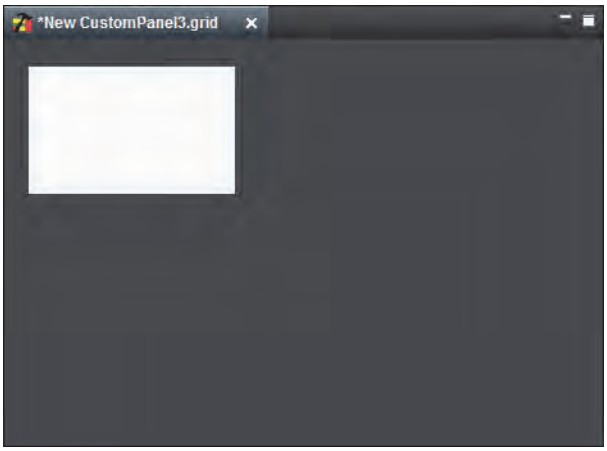



***Figure 5.29*** *- Component anchored to edge of panel*

If the panel is shrunk in size, the blank canvas component remains the same distance from the left and right edges of the panel while its size is shrunk horizontally but not vertically, as demonstrated in **Figure 5.30**.

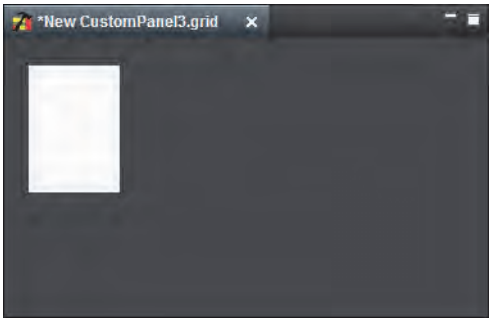
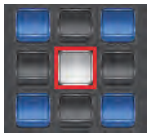



***Figure 5.30*** *- Component anchored to edge while panel is expanded*

- Anchoring to the center of the canvas



Select the center quadrant from the Anchor Points/Background Alignment menu to anchor the component to all four corners of the canvas. When the canvas is expanded or shrunk, the component will remain a fixed distance from the edges and corners of the canvas according to the set pixel amounts for the top, left, bottom, and right. If the canvas size is adjusted vertically or horizontally, the component height and width size will adjust accordingly while maintaining fixed offsets from the top, left, bottom, and right.

For example, the blank canvas component in **Figure 5.31** is anchored to the center of the canvas.

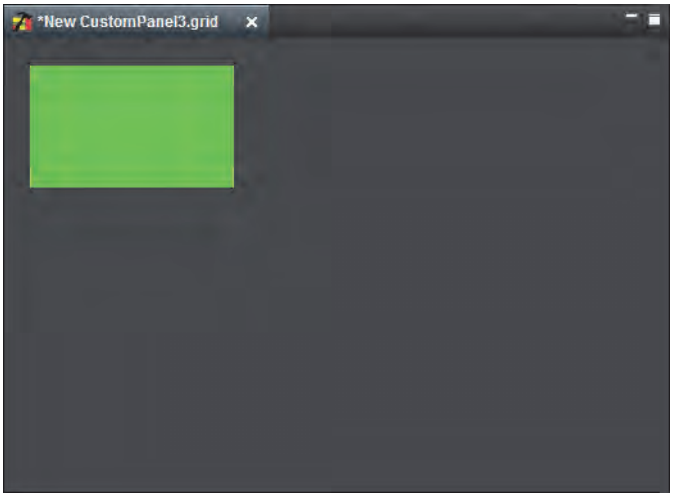



***Figure 5.31*** *- Component anchored to center of panel*

If the panel is resized, the blank canvas component remains in the same position from all edges of the panel but shrinks horizontally and vertically as the panel is shrunk, as demonstrated in **Figure 5.32**.

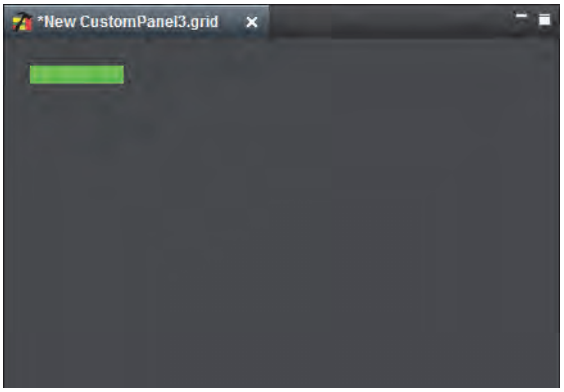



***Figure 5.32*** *- Component anchored to center while panel is expanded*

## Locking Panel Proportions

By locking or unlocking proportions, you can control how a CustomPanel is displayed when the DashBoard window is resized.

**To lock or unlock proportions**

- While in edit mode, right-click an empty space on the panel and then select the available option:
  - **Lock all proportions** — Components in the panel automatically resize to fit the new size and shape of the window.

    This option is useful for accommodating different screen sizes and resolutions.
  - **Unlock all proportions** — Components maintain their current sizes and shapes.

    This option is useful for ensuring a consistent visual display.

## Parameters and Data Sources

The configuration and state of any DashBoard Connect device is represented by a set of parameters. Device parameter data can be edited to change device settings. These parameters appear in the DashBoard Device View as various components and can be dragged into your CustomPanel.

You can also define new local parameters for your CustomPanel, and reference them in scripts. If you want to share these parameters with other CustomPanels or pass information to other Ross products such as XPression, you must create an XML data file and associate it with your CustomPanel.

When you create a CustomPanel, you can opt to automatically create an XML data file to store data for parameters you create in PanelBuilder. Alternatively, you can create a blank self-contained data source panel. For more information about creating a new CustomPanel, see "**Creating a CustomPanel**" on page 5–8.

Data sources contain parameter data which can be displayed and/or manipulated in a CustomPanel. A data source can be an XML data file, an openGear configuration file (.ogd), or a device.

Underlying every CustomPanel is a hierarchy of component elements, each of which can be associated with only one data source. Data source scope cascades. If no data source is specified for a given element in the hierarchy it inherits its data source association from its parent element. To view the element hierarchy, enter Edit Mode, double-click an element, and look at the component hierarchy tree in the top left portion of the Edit Component window.

When you associate a CustomPanel with a data source, the parameter library of the data source is inherited by the CustomPanel. If you had components on a CustomPanel before associating the panel with a data source, those original components will now have access to the same data source library as any new ones you add. You can this library to further customize your CustomPanel using OGP tags.

**Note:** If you do not need to pass information to other CustomPanels or applications such as XPression, and only need status summary information from other openGear devices, you do not need to associate a data source with your CustomPanel.

### The Add/Edit Parameter Window

You can create and modify parameters in the Add/Edit Parameter window. The Add/Edit Parameter window appears when you click the **Parameters** button on the **Edit Mode** toolbar.

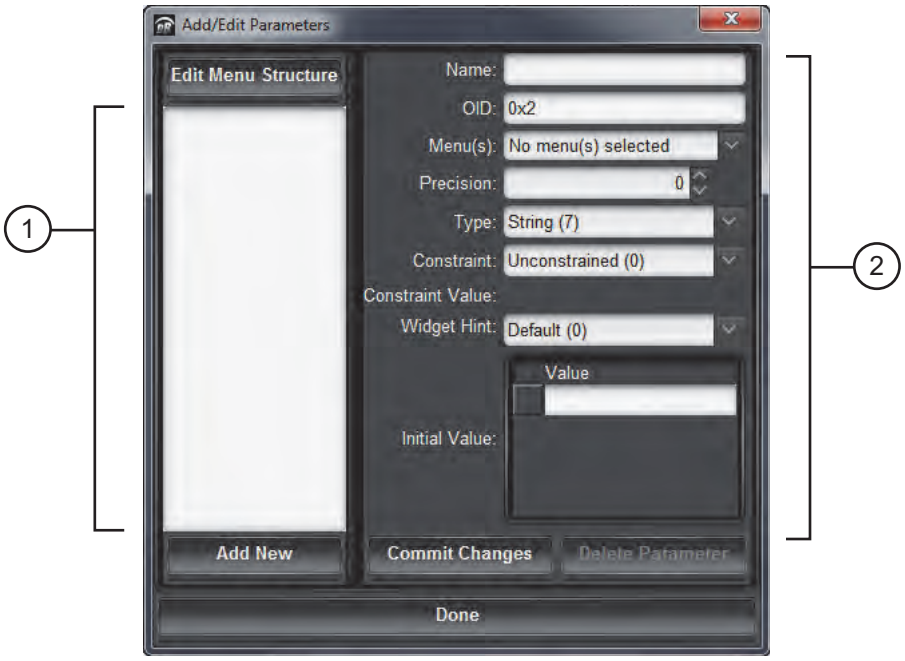



***Figure 5.33*** *- The Add/Edit Parameter Dialog*

The Add/Edit Parameter dialog provides the following information:

### 1. List of Parameters

Provides a list of parameters, arranged according to the menu structure and assigned OID tags, currently available for the selected CustomPanel component. Parameters can be created by any of the following:

- an associated data source
- tools in the **Edit Mode** toolbar
- a component dragged from a device in the Tree View of DashBoard

You can select a parameter from the list to display its information in the window and to edit its properties. You can also edit the menu structure.

### 2. Parameter Information

This area enables you to quickly view and/or modify the properties of a selected parameter.

## Creating a New Parameter

You can create new parameters using the Add/Edit Parameter dialog. Once saved, the parameters are available to be referenced in scripts and when creating data-backed objects in the CustomPanel.

You can also edit the parameter menu structure. The parameter menu structure only applies to panels that have an external XML data source, and not to panels that have a self-contained data source.

#### To define a new parameter

1. On the **Edit Mode** toolbar, click the **Parameters** button.

   The **Add/Edit Parameter** dialog appears.

2. Click **Add New**.

3. In the **Name** box, type a unique name for the parameter. The parameter will be identified in other dialogs using this name, so ensure to create a descriptive name.

4. In the **OID** box, type an object ID. Each parameter must be identified by a unique object identifier.

   **Tip:** A unique two-byte hexadecimal OID is automatically provided when the **Add New** button is selected. You can accept this OID or change it. OIDs do not have to be hexadecimal values. They are string data.

5. In the **Menu(s)** list, select the menu you want the parameter to appear under.

6. Use the **Precision** field to define the number of digits following the decimal point displayed for printed numbers. It applies mainly to floating point (float) numbers.

7. Specify the storage type for the parameter value using the **Type** menu. Choose from the following:

   - **String** — Specifies that the parameter value is an alpha-numeric series of characters (can be text or a mix of text and numbers).
   - **String Array** — Specifies that the parameter can contain multiple string values.
   - **Integer** — Specifies that the parameter value is a number without decimal places. Select 16 or 32 bit.
   - **Integer Array** — Specifies that the parameter can contain multiple integer values. Select 16 or 32 bit.
   - **Float** — Specifies that the parameter value is a floating-point number (uses decimals) or a number with an exponent.
   - **Float Array** — Specifies that the parameter can contain multiple float values.

8. Specify additional limitations on the parameter values using the **Constraint** menu. Note that the available options depends on what you selected in step 7. Choose from the following:

   - **Unconstrained** — Select this option when using a string type. No limitations are applied to the parameter value. For example, a text field parameter where a user can type any word or mix of letters and numbers.
   - **Range Constraint** — Select this option when using an integer or float type. Use this option to stipulate a range of numbers that the user can select from (e.g. minimum and maximum values). For example, use to stipulate a range from 1-10.
   - **Choice Constraint** — Select this option when using a string type to provide a specific list of options (1, 2, 3, or 4) the user chooses from where each option is associated with a tag.
   - **Alarm Table** — select this to set constraint values for alarm states.
   - **String Choice (for table widget only)** — select this when using a string type to provide a specific list of strings from which the user chooses.

9. If you did not select **Unconstrained**, use the **Constraint Value** area to define the valid set of values for the parameter:

   - For choice constraints, including string choice, do the following once for each valid value:
     - › In the **Value** column, click **[insert value]**, type a valid value, and then press **Enter**.
     - › In the **Name** column, type a name for the value.

       The name is associated with the parameter value, and appears on labels, etc.

       **Note:** The **Name** column is available only if the parameter is a numeric type.
   - For range constraints:
     - › In the **Minimum** box, type the lowest valid value.
     - › In the **Maximum** box, type the highest valid value.
     - › In the **Step Size** column, type the step size.

       For example, if valid values must be evenly divisible by 10, type 10.
     - › If you plan to use a touch wheel in your panel, select the **Loop** check box.
   - For alarm table constraints, do the following once for each valid value:
     - › In the **Bit** box, type the bit value for the constraint value. For example, you may have two options; 1 and 0. You would have one row for each bit state.

       The bit must be unique for each constraint value.
     - › In the **Severity** box, select a severity level.
     - › In the **String** box, type the alarm message you want associated with this constraint value.

10. Specify the graphical display hint for the parameter using the **Widget Hint** menu. Choose from the following:

    - **Default** — Displays the parameter as defined according the data source.
    - **Read-only text** — Displays the parameter as a status text field that cannot be altered by the user. A border and background is automatically applied to the field.
    - **Label** — Displays the parameter as a text field without a border or background.
    - **Text Entry** — Displays the parameter as a single line text field that is editable by the user. The user must enter one of the values defined using the **Constraint Value** field.
    - **Multi-Line Text Entry** — Displays the parameter as a text field with more than one line. The user must enter one of the values defined using the **Constraint Value** field.
    - **HTML Content** — Displays the parameter as a field that requires the user to input HTML data.
    - **Editable Dropdown List** — Displays the parameter as a menu that the user clicks to display an expanded list of values to choose from. These values are determined by the **Constraint Value** field.
    - **Alarm-Style Colored Dot** — Displays the parameter as a status indicator, similar to an LED, that updates based on conditions defined in the **Constraint Value** field.

11. In the **Initial Value** area, specify the initial value for the parameter. If the parameter is an array, you can specify multiple values.

12. Click **Commit Changes** to save your new parameter.
13. Click **Done** to exit the dialog.

**To edit the parameter menu structure**

1. On the **Edit Mode** toolbar, click the **Parameters** button.

   The **Add/Edit Parameter** dialog appears.
2. Click **Edit Menu Structure**.

   The **Add/Edit Menus** dialog appears.
3. In the list on the left, click the folder under which you want the new menu to appear (Status or Config).
4. In the **Menu Name** box, type a name for the menu.
5. Click **Insert Menu**.
6. If you want to delete a menu, click it in the list on the left, and then click **Delete Menu**.
7. When you are satisfied with the menu structure, click **Save Menu**.
8. Click **Done**.

## Associating a Data Source with a CustomPanel

You can associate a specific data source, such as an openGear card or other DashBoard Connect devices, or a saved configuration file (*.ogd) with a CustomPanel. Doing so enables you to use that data source as a library of parameters which you can then manipulate. Additional tools in the CustomPanel's Edit Mode toolbar enable you to quickly select components based on a specific device. Which components are available depends on the specific device you have selected. For example, the components of an UDC-8625A-B differ from those of an XPression™ Real-time Motion Graphics System. We highly recommended you become familiar with the device(s) before using this PanelBuilder feature.

### Associating a Data File with a CustomPanel

Associating a saved configuration file (*.ogd) enables you to use the parameters saved in that file as a type of library for the CustomPanel. Once you have successfully associated a data file to the CustomPanel, the Edit Mode toolbar automatically updates to display the new tool options. You can also use a data file as a base for creating an interface in DashBoard that auto-populates fields in a Ross XPression system, or to familiarize yourself with PanelBuilder features without directly impacting a device.

**To associate a data file with a CustomPanel**

1. Select from the **Edit Mode** toolbar.
2. Double-click the CustomPanel area. The border displays the **Edit Component: <abs>** dialog when the mouse button is released.
3. Select the **Abs Attributes** tab.
4. In the **Device Control** area, select the **openGear** check box.
5. Click **configure** to display the **Select Device for Context** dialog.
6. Click **Select a file**.
7. Click **Browse...** to display the **Select Device Data Source** dialog.
8. Navigate to the required *.ogd file.
9. Click **Select Device Data Source** to close the dialog and update the **File** field.
10. Click **OK**.
11. Click **Apply Changes** or **Apply and Close**.

## Router Control Panels

You can create a CustomPanel to control Ross Video NK Series video routers. PanelBuilder includes special tools for making the creation of a router control panel easy.

For comprehensive information about NK Series routers and Internet Protocol Servers, consult the user manuals that came with your NK Series router system.

Router control panels created in PanelBuilder typically include the following components:

- A list of Internet Protocol Server (IPS) selectors. An IPS is a device that controls one or more routers. An IPS selector list enables you to choose a set of routers to control.
- A list of levels. Levels are logical groups of inputs and outputs. Levels ensure that a certain set of inputs can only be routed to a certain set of outputs. This is useful in facilities that deal with multiple video formats, to prevent routing of one signal format to a device that expects to receive a different format.
- One or more lists of sources. Sources are video router inputs.
- One or more lists of destinations. Destinations are video router outputs.
- A group of functions. Functions are router commands, such as chop, take, configure, etc.
- A level status table. The level status table lists levels and the sources and destinations associated with them.

To create a router control panel, you create a CustomPanel, then add the components listed above.

**Interface Design Tips:**

- Do not place lists of sources, destinations, or levels in the same container component as an IPS, or in a higher-level container. These lists must reference an IPS that is at a higher level in the component hierarchy.
- Place lists of sources, destinations, or levels in container components that include scrollbars. This is important because these lists usually have a lot of buttons.
- By default, the buttons in a list are all in one row. If there are a lot of buttons, this may not be a usable interface. You can create a table with the maximum elements per row specified, and then edit the panel source to move the list into a row of the table. The buttons are then displayed as a matrix.

  You can achieve the same effect by creating **`a simplegri`**d tag in the source, instead of a table.

**To create a CustomPanel for use as a router control panel:**

1. Create a new CustomPanel.

   When prompted to specify the panel type, choose **Blank Canvas**.

   For information about how to create a new panel, see "**Creating a CustomPanel**" on page 5–8.

2. Press **Ctrl+G** to enter edit mode.

   Grid lines appear on the panel.

3. Double-click the panel to open the **Edit Component** window.

4. On the **ABS Attributes** tab, in the **Data Source/Device Control** area. select the **NK Series Routers** check box.

   The **Select IPS** dialog appears.

5. Click the IPS you want to associate with your router control panel, and then click **OK**.

6. Click **Apply and Close**.

   The Edit Mode toolbar now includes buttons for adding router control components, as shown in **Figure 5.34**.

***Figure 5.34*** *- Edit Mode Toolbar Including NK Buttons for Adding NK Router Controls*

**To Add a list of IPS selectors:**

1. On the **Edit Mode** toolbar, click the **NK IPS** button.

   **Tip:** If the **NK IPS** button is not visible, click the **NK NK** button to reveal the **NK IPS** button.

2. Drag a box on the panel to define the list area.

   A row of buttons appears. Each button represents one IPS, which can be selected by the user.

**To insert a list of destinations:**

1. Create a container component, such as a blank canvas, to contain the list.

   **Note:** Do not place lists of destinations in the same container component as an IPS selector, or in a higher-level container. These lists must reference an IPS selector that is at a higher level in the component hierarchy.

2. On the **Edit Mode** toolbar, click the **NK DST** button.

   **Tip:** If the **NK DST** button is not visible, click the **NK NK** button to reveal the **NK DST** button.

3. Drag a box on the panel to define the list area.

   The **Insert into Component** dialog appears.

4. Do one of the following, and then click **Ok**:

   - If you want to include all destinations in the list, select the **Show all (dynamic)** check box.
   - If you want to include only some destinations, deselect the **Show all (dynamic)** check box, and then select the individual destinations you want to include in the list.

   The destination buttons appear.

**To insert a list of sources:**

1. Create a container component, such as a blank canvas, to contain the list.

   **Note:** Do not place lists of sources in the same container component as an IPS selector, or in a higher-level container. These lists must reference an IPS selector that is at a higher level in the component hierarchy.

2. On the **Edit Mode** toolbar, click the **NK SRC** button.

   **Tip:** If the **NK SRC** button is not visible, click the **NK NK** button to reveal the **NK SRC** button.

3. Drag a box on the panel to define the list area.

   The **Insert into Component** dialog appears.

4. Do one of the following, and then click **Ok**:

   - If you want to include all sources in the list, select the **Show all (dynamic)** check box.
   - If you want to include only some sources, deselect the **Show all (dynamic)** check box, and then select the individual sources you want to include in the list.

   The source buttons appear.

**To insert a list of levels:**

1. Create a container component, such as a blank canvas, to contain the list.

   **Note:** Do not place lists of levels in the same container component as an IPS selector, or in a higher-level container. These lists must reference an IPS selector that is at a higher level in the component hierarchy.

2. On the **Edit Mode** toolbar, click the **NK Level** button.

   **Tip:** If the **NK Level** button is not visible, click the **NK NK** button to reveal the **NK Level** button.

3. Drag a box on the panel to define the list area.

   The **Insert into Component** dialog appears.

4. Do one of the following, and then click **Ok**:

   - If you want to include all levels in the list, select the **Show all (dynamic)** check box.
   - If you want to include only some levels, deselect the **Show all (dynamic)** check box, and then select the individual levels you want to include in the list.

   The levels buttons appear.

**To insert a function button:**

1. On the **Edit Mode** toolbar, click the **NK Func** button.

   **Tip:** If the **NK Func** button is not visible, click the **NK NK** button to reveal the **NK Func** button.

   **Tip:** The **configure** function inserts a configure button in the panel. This enables users to open the Switchboard configuration dialog, where they can configure the sources, destinations, and levels assigned to a particular IPS.

2. Drag a box on the panel to define the button area.

   The function button appears.

**To insert a level status table:**

1. On the **Edit Mode** toolbar, click the **Insert a level status table** button.
2. Drag a box on the panel to define the table area.

   The **Insert into Component** dialog appears.
3. In the Status Table Type area, select one of the following:
   - **Show status for all destinations**
   - **Show status for currently selected destination**
4. From the scrolling list, select a scrolling option:
   - **True** — use vertical and/or horizontal scroll bar, if needed.
   - **False** — do not use scroll bars.
   - **Vertical** — add a vertical scroll bar.
   - **Horizontal** — add a horizontal scroll bar.
   - **Always** — always use vertical and horizontal scroll bars.
5. Click **Ok**.

   The level status table appears.

# DataSafe™

This chapter provides instructions for configuring and using the DataSafe™ feature in DashBoard. Refer to the ***MFC-8300 Series User Manual*** for details on setting up your MFC-8300 series Network Controller Card.

The following topics are discussed:

- DataSafe Overview
- DataSafe Basics

## DataSafe Overview

DataSafe enables you to save openGear card parameters to a file, and later restore those parameters to one or more cards of the same type. Multiple configuration sets can be stored if required. This gives you the flexibility of configuring a large number of cards identically from a single stored configuration. This feature is enabled or disabled on a slot-by-slot basis and currently defaults to OFF.

This feature is available for frames using the MFC-8310-N and MFC-8320-N series cards only. Refer to your openGear card manual for more information on using DataSafe with your card.

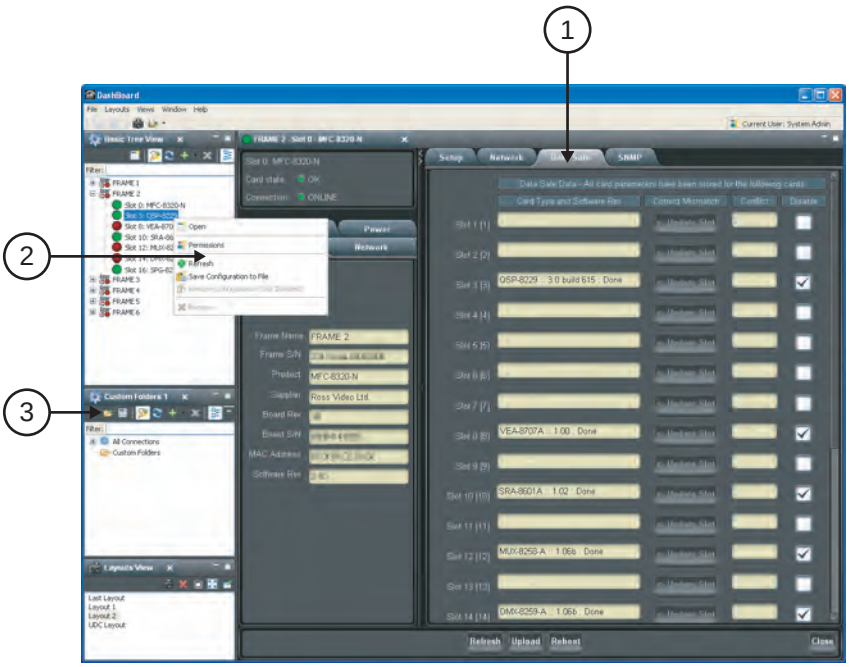



***Figure 6.1*** *DataSafe Tab Overview*

### 1. DataSafe Tab

The **DataSafe** tab is available in the **Device Editor** of a MFC-8310-N series or MFC-8320-N series Network Controller Card. The tab lists the cards installed in the frame in slot ascending order. The following items are displayed in the tab:

- **Slot Name** — The slot name is set in the **Card Slots Names** field of the **Setup** tab. The physical slot that the card is installed in is indicated by the number in brackets after the Slot Name. For more information on re-naming slots, refer to the section "**Re-naming an openGear Slot in the Tree View**" on page 7–2.
- **Card Type and Software Rev** fields — This area indicates the card installed in the slot and the software version. The information is displayed in the form of **xxxx :: #### :: y** where **xxxx** represents the card type, **####** represents the software revision, and **y** represents the card status. If the field is blank, there is no saved data for

this slot. For example, in Slot 3 of **Figure 6.1**, the QSP-8229 is running with software version 3.0 build 615 while Slot 2 does not have a card installed (therefore the field is blank).

- **Correct Mismatch** fields — This area includes the **Update Slot** button. Clicking this button automatically updates the slot with the new card information indicated in the **Conflict** field.
- **Conflict** fields — This field is blank if the card currently in the slot has the same card type and software version as the saved DataSafe data. When the software version or the card type do not match the saved data for the slot, the **Conflict** field for that slot is populated with the new information, alerting the user that a new configuration is available for that slot. The information is displayed in the form of **xxxx :: #### :: y** where **xxxx** represents the card type, **####** represents the software revision, and **y** represents the card status. If the field is blank, there is no saved data for this slot.
- **Disable** check box — This option disables the **DataSafe** feature for that slot.
- **Force** button — If any slots have a software version mismatch, the **Force** button allows the user to load the current DataSafe data to the cards and overwrites the frame-stored data with the card's current settings. This button is enabled only if a software version incompatibility exists. For more information on forcing DashBoard to update cards with software incompatibilities, refer to the section "**Forcing DataSafe Updates**" on page 6–5.
- **Mask Warning** check box — The MFC-8300 series Network Controller card displays a warning in the **Data Safe State** field of the **Hardware** tab if any of the card slots have an error. For example, if a card is installed that does not match the last saved data for that slot. If the check box is selected, no warnings are displayed in DashBoard. The default setting is to display warnings and errors.

### 2. Save/Restore Device Parameters in the Basic Tree View

The following DataSafe options may be available when you right-click a device in the Tree View:

- **Open** — Selecting this option enables you to display a Device Editor tab for the selected device.
- **Save Configuration to File** — Selecting this option displays the **Export Slot to file** dialog from which you can save a DataSafe file for the specific card on your DashBoard host machine. For more information on saving DataSafe files, refer to the section "**Saving a DataSafe File**" on page 6–3.
- **Restore Configuration** — Selecting this option displays the **Restore Configuration Wizard** from which you can restore from a DataSafe file for the specific card(s) on your DashBoard host machine. For more information on recalling configurations from a file, refer to the section "**Restoring Configurations to Devices**" on page 6–3.

### 3. Save/Restore Configurations for Devices in the Advanced Tree View

With the Advanced Tree View feature, the following options are available in the Custom Folder toolbar:

- **Save Configuration to File** — Selecting this option displays the **Export to file** dialog from which you can save the configuration of all the devices in the Custom Folder View to a single DataSafe file.
- **Restore Configuration** — Selecting this option displays the **Configuration Wizard** from which you can recall all configurations for all devices in the Custom Folder View from a ***.tvc** file.

## DataSafe Basics

This section provides instructions for saving and recalling DataSafe files. DashBoard saves all card parameter values to the file which is stored locally on the DashBoard client host machine.

✶ Some openGear cards have card-edge jumper settings that disable remote control. Ensure that your card is configured to enable control from DashBoard before saving and recalling DataSafe files.

## Saving a DataSafe File

DashBoard enables you to save card parameters to a unique DataSafe file which you can name and store on your computer. If you are using the **Advanced Tree View**, an option also exists that enables you to save the configuration of all devices in the specified custom folder. Both methods are described in this section.

### Saving a DataSafe File for a Specific Device

This section outlines how to save a DataSafe file containing all of the data of a specific card including the card parameters and settings, frame name, slot name and number, card type, and software revision. This option is always available regardless of which MFC-8300 series Network Controller card is used and which software options have been purchased.

**To save a DataSafe file for a specific device:**

1. From the **Tree View**, right-click the card you wish to save the configuration for.
2. Select to display the **Export to file** dialog box.
3. Navigate to the location you wish to save the file to.
4. Enter a filename in the **File name:** field.
5. Ensure the **Save as type:** field is set to **openGear Device File** (***.ogd**).
6. Click **Save**.

### Saving a DataSafe File for a Group of Devices

This section outlines how to save the configuration of devices in the Custom Folder View to a single DataSafe file. This file is in the format ***.tvc** (Tree View Configuration) and is stored on the DashBoard host machine.

**To save a DataSafe file for a group of devices:**

1. From the **Advanced Tree View**, select the Custom Folder to save the configuration for.
2. Select to display the **Export to file** dialog box.
3. Navigate to the location you wish to save the file to.
4. Enter a filename in the **File name:** field.
5. Ensure the **Save as type:** field is set to **Tree View Configuration** (***.tvc**).
6. Click **Save**.

## Restoring Configurations to Devices

DashBoard can send a DataSafe file to a specific MFC-8300 Network Controller Card and slot for use. Once received, the Network Controller Card updates the card if the installed card matches the data sent. You can also choose to copy parameters from a card to a single device or group of devices. Both methods are described in this section.

As part of the recall process, DashBoard opens a DataSafe file, examines the card type and software revision, and determines what devices on the network in the current Tree View that match. The user can then select which card(s) to update with the new values.

✶ You must have an MFC-8310-N series or MFC-8320-N series installed in the frame to recall DataSafe data or files.

### Restoring a DataSafe File

This section outlines how to recall a saved DataSafe file on your DashBoard host machine to a single type of device your network.

**To recall a DataSafe file:**

1. From the **Tree View**, right-click the device to update.
2. Select to display the **Restore Configuration Wizard** dialog box.
3. From the **Configuration Source** list, select **Load Parameters from a File**.
4. Click **Next >** to display the **Select Device File** menu.
5. In the **Configuration File** field, enter the DataSafe file to download to your card or click **Browse...** to navigate to the file location on your DashBoard host machine.
   - Ensure that the select file is a valid ***.ogd** file.
   - Information, such as the card type, slot and frame names, is displayed in the **Device Info:** field. An error message will also display in this field if a card type or software version mismatch will occur.
6. Click **Next >** to display the **Select Destination** menu. The **Select Destination** menu provides a list of the compatible cards based on the card selected in step 1
7. Select the device(s) to recall the file to:
   - From the **Select Destination** list, select the check box(es) for the devices you wish to recall the file to.
   - Verify that the device(s) you wish to recall the file to. The **Error/Warning** fields indicate any errors, such as incompatible software or card type mismatch.
8. Click **Finish**.

### Restoring Parameters for a Group of Devices

You can recall all configurations for all devices in the Advanced Tree View from a ***.tvc** file. Card parameters will only load from the DataSafe file if the card type matches and the software versions are compatible.

**To recall parameters for a group of devices in the Advanced Tree View:**

1. From the **Advanced Tree View**, select the Custom Folder to recall the configuration file for.
2. Select from the Custom Folders toolbar to display the **Restore Configuration Wizard**.
3. From the **Configuration Source** field, select **Load Parameters from a File** to display the **Select Device File** menu.
4. To enter a filename in the **Configuration File:** field:
   - Click **Browse...** to navigate to the file location on your DashBoard host machine.
   - Ensure that the select file is a valid ***.tvc** file.
   - Information, such as the card type, slot and frame names, is displayed in the **Device Info:** field. An error message will also display in this field if a card type or software version mismatch will occur.
5. Click **Next >** to display the **Select Destination** menu. The **Select Destination** menu provides a list of the compatible cards based on the card selected in step 1
6. Select the devices to restore as follows:
   - From the **Select Destination** list, select the check box(es) for the devices you wish to recall the file to.
   - The **Error/Warning** fields indicate any errors such as incompatible software or a card type mismatch.
7. Click **Finish**.

### Copying Parameters

**To copy parameters from one device to another:**

1. From the **Tree View**, right-click the device to update.
2. Select to display the **Restore Configuration Wizard** dialog box.
3. From the **Configuration Source**, select **Copy Parameters from another Card**.

4. Click **Next >** to display the **Select Source Device** menu.
5. From the provided list, select the card whose settings will be copied.
6. Click **Next >** to display the **Select Destination** menu.
7. From the provided list, select the device(s) to copy the parameters to. The **Error/Warning** fields indicate errors such as software or card type mismatch.
8. Click **Finish**.

## Notes on Saving and Restoring Parameters

This section provides brief operational notes when saving and restoring parameters:

- The **Restore Configuration Wizard** dialog box includes the **Hide invalid destinations** check box. Selecting this check box hides cards in the **Select Destination** list that are not applicable based on the card type you selected to restore and the software of that card. Only those cards that are the same type and compatible software versions will be listed.
- The **Select Destination** list displays the frame and cards in the same format as seen in the Tree View. For example, if the frame node is expanded in the Tree View, the same node is expanded in the **Select Destination** list.
- You can select all devices in a single frame by selecting the checkbox beside the frame entry in the **Select Destination** list.
- The ▣ indicator beside the frame name in the **Select Destination** list indicates that at least one device in that frame is selected. A checkbox beside the frame name indicates that all the devices in that frame are selected.

## Forcing DataSafe Updates

If any slots have a software version mismatch, the **Force** button allows the user to have the current DataSafe data loaded to any slots where a software mismatch is occurring. This button and text only displays if a software version mismatch occurs in the DataSafe tab.

A mismatch is reported in the **Conflict** field of a slot if a new card is installed into that slot is a different card type, or has a software incompatible version, than the current DataSafe file. If this new card remains installed into that slot for more than approximately 24 hours, then the parameters of the new card will automatically be saved as the new DataSafe file.

The following tasks are performed when the **Force** button is clicked:

- Loads all of the data from the card
- Stores the data as the new DataSafe data
- Clears the information in the **Conflict** field
- Updates the Slot field in the **DataSafe** tab

# Configuring Devices

There are two basic methods to configuring devices: offline and online. An online device is one that is actively communicating with your DashBoard client. You make changes in the Device Editor tab that take affect immediately on that device. An offline device is one whose configuration has been saved to a file for offline configuration. In this case, you use the File Navigator tab to edit a device configuration file that can then be saved on your computer. Changes made to this file do not affect the actual device. This device configuration file can then be applied to other devices of the same type. This chapter provides information on both methods.

The following topics are discussed:

- Configuring Online Devices in DashBoard
- Using the File Navigator
- Upgrading Device Software

## Configuring Online Devices in DashBoard

DashBoard enables you to configure devices in real-time. Each device has specific configuration parameters, depending on the device you have selected in the **Tree View**. For example, you may wish to change a specific parameter on a device while it is online in your openGear frame, or re-configure a device, or upload new device software when it becomes available.

✶ Using the **Reboot** button takes the device off-air during the reboot cycle.

### Configuring Devices in DashBoard

**To configure and verify device information in DashBoard:**

1. From the **Tree View**, double-click a device to display a corresponding **Device Editor** tab. In the following example, the **Device View** displays settings for the QSP-8229 located in Slot 3 of FRAME 2.

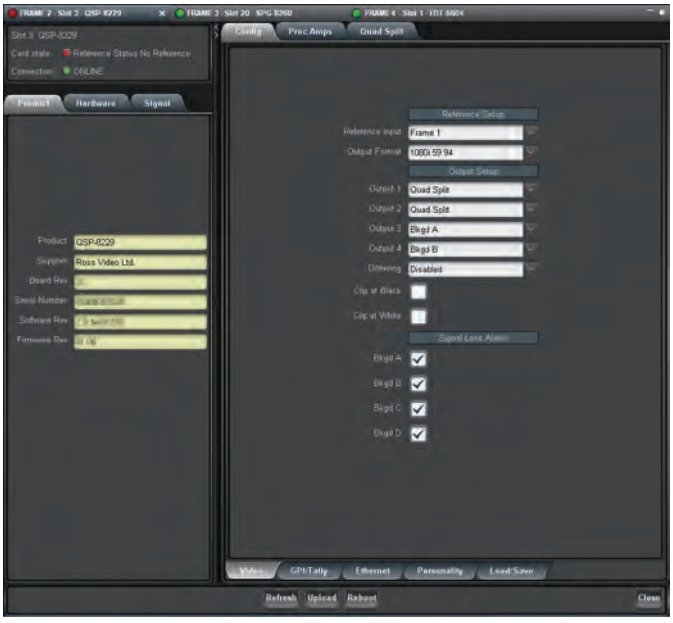



***Figure 7.1*** *Device Tab Example*

2. Configure the required parameters using the controls provided in the **Settings Area**, such as those in the **Setup** tab seen in the previous example. Refer to the manual for your device for information on available parameters and menus.

## Re-naming an openGear Slot in the **Tree View**

DashBoard offers two methods to re-name a card slot. Both methods are described in this section.

- **Using the Setup tab of the Network Controller Card** — Use this method to rename the card and have the change effective for the network. The new name displays in all DashBoard Tree Views for all DashBoard workstations connected to that card.
- **Using the Rename option for a Custom Subfolder item** — Use this method to rename the card for your local DashBoard workstation only. Other DashBoard workstations will not display the new name.

### Using the Setup tab

**To re-name an openGear card using the Setup tab:**

1. From the **Tree View**, double-click the MFC-8300 Network Controller Card to display a corresponding **Device Editor** tab.
2. Select the **Setup** tab.
3. In the **Card Slot Names** section of the **Setup** tab, locate the slot you wish to re-name.
4. Enter the new name for the card slot in the text field provided. The new name displays in all instances of the card for all DashBoard workstations.

### Using the Rename Option

**To re-name a device using the Rename option:**

1. From the **Custom Folder** in the **Advanced Tree View**, right-click the device you wish to re-name.
2. Select to display the **Rename** dialog box.
3. Enter the new name for the device in the **Name:** field.
4. Click **OK**. The new name is only displayed on your local DashBoard workstation.

## Automatic Discovery

The Automatic Discovery feature is enabled by default, and allows DashBoard to automatically search for new devices. You can specify that DashBoard search all interfaces on the same network, or search only those interfaces you have selected.

**To configure the Automatic Discovery feature:**

1. From the main toolbar, select **Window > Preferences**. The **Preferences** dialog box opens.
2. Select **Automatic Discovery** to display the **Automatic Discovery** dialog box.

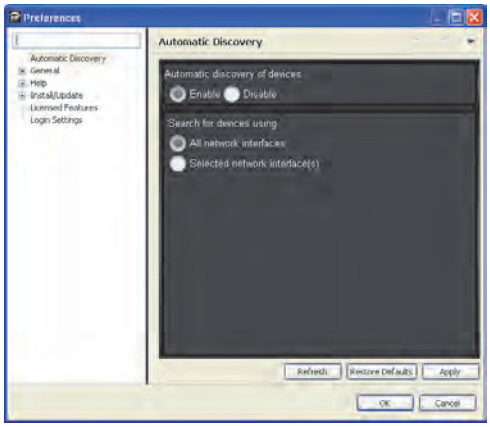



***Figure 7.2** Automatic Discovery Dialog Box*

3. Select the **Enable** check box in the **Automatic Discovery of devices** section to enable the feature. Deselecting the check box disables the Automatic Discovery feature.

4. Select a **Discovery** mode as follows:
   - **All network interfaces** — Select this option to enable DashBoard to find openGear cards only on the local network. DashBoard queries the network every 10 seconds and display new devices in the **Tree View**s. This is the default setting.
   - **Selected network interface(s)** — Select this option to enable DashBoard to query the specified network every 10 seconds and display new devices in the **Tree Views**.
5. Click **Apply**.

### Troubleshooting

If you are unable to make changes to the parameters of a device, or the **Upload** or **Reboot** buttons are disabled:

- verify that any edit permissions for the device are enabled in the **Device** tab.
- verify any relevant controls on your device hardware are set to enable remote control.

Not all openGear devices support the edit permissions feature and it is recommended that you refer to the documentation for your device for details. This control typically appears on the **Setup** tab of the **Device View**

### Removing Devices from the Tree View

If a device is still listed in the **Tree View**, but with a grayed out status indicator, then DashBoard is no longer communicating with the device. You can remove such devices from the **Tree View** using the following procedure.

**To remove all offline devices in an openGear frame:**

1. In the **Tree View**, right-click the frame containing offline devices.
2. Select **Remove offline devices** to remove the offline devices from the **Tree View**.

## Using the File Navigator

The **File Navigator** tab in DashBoard enables you to quickly navigate, manage, and update device configuration files. A device configuration file stores the configuration of a specific device. This functions much like the DataSafe feature for openGear cards where you can save the settings of one device to a file and recall that same file to a device of the same model. The File Navigator enables you to edit the settings of a device separate from the actual device, allowing you to store the new configuration as a separate file or update the current one.

Just like the other tree views, the File Navigator tab displays device configuration files in a hierarchy. But the hierarchy is determined by the folder organization on your computer. Note that when you collapse a folder in the File Navigator tab, every device file under the folder is disconnected.

**To display a File Navigator tab:**

1. Select **Views** from the main toolbar.
2. Select **File Navigator** to display a new File Navigator tab in the DashBoard window.

**To display your device files in a File Navigator tab:**

1. Click on the **File Navigator** toolbar to display the **Browse For Folder** dialog box.
2. Navigate to the folder on your computer where your device configuration files are stored.
3. Select **OK** to close the dialog box and update the tree view in the **File Navigator** tab. The selected folder is displayed.

**To edit a device file:**

1. In the tree view of the **File Navigator** tab, expand the node of the folder to view a list of available device configuration files. You can also search the file hierarchy by entering text in the **Filter** field.
2. Double-click the file you wish to edit to establish a connection to it. The icon will no longer be grayed-out.

3. Display a Device Editor tab in the DashBoard window.
4. Configure the device parameters as required. An asterisk (*) displays on the File Navigator tab and on the applicable Device Editor tab whenever there are unsaved changes in a device configuration file. The file that has unsaved changes is also set in blue in the **File Navigator** tab.
5. Save your changes using one of the options in the **File** menu. You can also use the provided buttons on the DashBoard main toolbar.

**To remove a device file from the File Navigator tab:**

1. In the tree view of the **File Navigator** tab, select the file you wish to remove. Note that this does not delete the file from your computer.
2. Click on the **File Navigator** toolbar.
3. Click **OK**.

**To refresh a directory:**

1. Right-click the folder in the tree view of the **File Navigator** tab.
2. Click **Reload**.

# Upgrading Device Software

DashBoard enables you to upload software updates to multiple devices available in the **Tree View**. You can select any number of similar devices to upgrade (upload software files to) and monitor the upgrade progress. This section summarizes the upload software process. The process may differ for your specific device. Refer to your device user manual for information on upgrading the software before proceeding.

## Upgrading Device Software

**To upload software to a device:**

1. Contact your Ross Technical Support representative for the latest software version for your device.
2. Display the **Device** tab of the device you wish to upload software to.
3. In the **Device** tab, click **Upload** to display the **Select File Upload** menu of the **Upload Software Wizard**.
4. Select a file to upload as follows:
   - Click **Browse...** to navigate to the *.bin upload file you wish to upload. DashBoard automatically selects the last directory that you loaded a file from.
   - Select a file to upload.
   - Select **Open** to return to the **Select File Upload** menu.
   - The **Select File Upload** dialog box now displays the path to the selected file, and information on the selected file such as name, type, load size, and creation date. A warning is displayed in the **Warnings** field when software conflicts occur, such as the selected file will downgrade the selected device.
5. Click **Next >** to display the **Select Destination** menu.
6. Select the device(s) you wish to upload the selected software file to as follows:
   - Note that only the device you selected in is selected.
   - If you wish to include other devices, select the desired devices from the **Select** drop-down list using one of the following options:
     - **Select All** — This option selects all the similar devices. The selected software file will be uploaded to all devices on the network.
     - **Select Without Warnings** — This option selects only those devices that do not include a conflict with the selected software file.

› **Select Without Warnings (include downgrades)** — This option uploads the software to similar devices that do not have any conflicts, but will include those devices that will be downgraded if the selected software file is uploaded to them.

› **Select None** — This option clears all the check boxes in the **Device** field.

› Click **Select**.

- The **Device** field, located below the **Select** drop-down list, displays information such as the device name and slot information, frame it is installed in, the current software version, and any applicable error messages are displayed.
- You can also select the devices to upgrade by selecting or clearing the corresponding check boxes in the **Device** field.
- If the **Hide invalid destinations** check box is selected, the menu only displays similar devices for the selected software file. For example, if you are attempting to upload a software file for a UDC-8625, only UDC-8625 cards will be displayed. If the check box is cleared, all devices currently installed on the network are displayed but are grayed out.

7. Click **Finish**. The **Uploading to Selected Devices** menu displays and the upload process begins. A progress bar displays in the **Uploading to Selected Devices** menu for each device you selected in 6

✶ Clicking during an upload can leave the device in an invalid state.

8. Monitor the upgrade process bar(s) displayed in the **Uploading to Selected Devices** menu while the software is upgraded to your device. You can also display the **Uploading to Selected Devices** menus as a new tab in DashBoard, allowing you to work in other DashBoard areas during the upload, by selecting the **Run in Background**.

9. When the upload is complete, click **OK** to close the **Uploading to Selected Devices** menu.

## Troubleshooting the Software Upload Process

Use the following information if the software upload process has failed:

- If the "**Selected file does not exist**" or "**Selected file is not a valid upload file**" error conditions are displayed in the **Upload Failed** dialog box, select **OK** from the dialog box and re-start the upload process and select the correct file.
- If a "**No response from device**" condition is encountered, the upload failed while in progress due to loss of power or communications. Verify that the device is powered on and that you have communication to the openGear frame. You must then restart the upload process.